

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
in this Office.

出 願 年 月 日
Date of Application:

1998年 4月28日

出 願 番 号
Application Number:

平成10年特許願第119099号

出 願 人
Applicant(s):

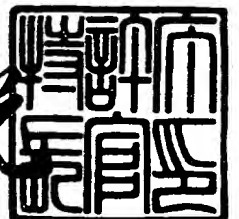
株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

1998年 5月29日

特 許 庁 長 官
Commissioner,
Patent Office

荒井寿光



出証番号 出証特平10-3042349

【書類名】 特許願

【整理番号】 11321201

【提出日】 平成10年 4月28日

【あて先】 特許庁長官殿

【国際特許分類】 G11C 7/00

【発明の名称】 不揮発性半導体メモリシステムの制御方法

【請求項の数】 14

【発明者】

 【住所又は居所】 神奈川県川崎市幸区堀川町 580 番 1 号 株式会社東芝
 半導体システム技術センター内

 【氏名】 田 中 義 幸

【発明者】

 【住所又は居所】 神奈川県川崎市幸区堀川町 580 番 1 号 株式会社東芝
 半導体システム技術センター内

 【氏名】 河 本 和 也

【特許出願人】

 【識別番号】 000003078

 【住所又は居所】 神奈川県川崎市幸区堀川町 72 番地

 【氏名又は名称】 株式会社 東 芝

【代理人】

 【識別番号】 100064285

 【弁理士】

 【氏名又は名称】 佐 藤 一 雄

【選任した代理人】

 【識別番号】 100073379

 【弁理士】

 【氏名又は名称】 佐 藤 政 光

【選任した代理人】

 【識別番号】 100088889

【弁理士】

【氏名又は名称】 橋 谷 英 俊

【選任した代理人】

【識別番号】 100082991

【弁理士】

【氏名又は名称】 佐 藤 泰 和

【先の出願に基づく優先権主張】

【出願番号】 平成 9年特許願第214561号

【出願日】 平成 9年 8月 8日

【手数料の表示】

【予納台帳番号】 004444

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9711285

【包括委任状番号】 9711282

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 不揮発性半導体メモリシステムの制御方法

【特許請求の範囲】

【請求項 1】

システムが管理する論理ブロックと、

複数のメモリセルにより構成され、前記論理ブロックに対応するデータを記憶する物理ブロックと、

前記物理ブロック中に含まれ、前記対応する論理ブロックのアドレスを記憶する冗長部と、

前記物理ブロックが少なくとも 2 以上で構成される物理ブロックエリアとを具備し、前記論理ブロックと前記物理ブロックエリアとの対応関係を管理するための論理番地／物理番地変換テーブルを作成する

ことを特徴とするメモリシステムの制御方法。

【請求項 2】

メモリアクセス時に、前記論理番地／物理番地変換テーブルを参照し前記論理ブロックに対応する物理ブロックエリアのアドレスを読み出し、前記物理ブロックエリアを構成する少なくとも 2 以上の物理ブロックの前記冗長部に記憶される対応する論理ブロックのアドレスを読み出すことにより、前記論理ブロックと対応する物理ブロックを選択することを特徴とする請求項 1 記載のメモリシステムの制御方法。

【請求項 3】

電源投入時に、前記論理番地／物理番地変換テーブルを作成することを特徴とする請求項 1 記載のメモリシステムの制御方法。

【請求項 4】

第 1 の所定単位によって管理されるファイルを、第 2 の所定単位に分割された記憶領域において記憶する半導体記憶装置において、

前記第 1 の所定単位の境界が、前記第 2 の所定単位の境界上に配置されるよう制御することを特徴とするメモリシステムの制御方法。

【請求項 5】

前記第 2 の所定単位は消去単位であることを特徴とする請求項 4 記載のメモリシステムの制御方法。

【請求項 6】

システムが管理するファイルと、
前記ファイルの内容を記憶するデータ領域と、
前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、
前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークし、対応する前記データ領域の消去を行うことを特徴とするメモリシステムの制御方法。

【請求項 7】

システムが管理するファイルと、
前記ファイルの内容を記憶するデータ領域と、
前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、
前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークしておき、メモリシステムに入力される信号に基づいて、前記管理領域の内容を検知し対応するデータ領域の消去を行うことを特徴とするメモリシステムの制御方法。

【請求項 8】

不揮発性半導体メモリシステムの制御方法において、セルアレイを複数個の物理ブロックに分割し、前記各物理ブロックにこのシステムが管理する前記論理ブロックとの対応付けの情報を記憶させ、

前記論理ブロックと前記物理ブロックとの、対応関係を管理するためのテーブルを前記システムの RAM 上に保持するに当り、

前記論理ブロックと前記物理ブロックの対応関係のうち、ホストからのアクセスに応じて必要な領域のみの対応関係を前記システム内の前記 RAM 上に逐次作成することを特徴とする不揮発性半導体メモリシステムの制御方法。

【請求項 9】

不揮発性半導体メモリシステムの制御方法において、セルアレイを複数個の物

理ブロックに分割し、前記各物理ブロックにこのシステムが管理する前記論理ブロックとの対応付けの情報を記憶させ、

前記論理ブロックと前記物理ブロックとの対応関係を管理するためのテーブルを前記システムのRAM上に保持するに当り、

前記論理ブロックと前記物理ブロックの対応関係のすべてを前記RAM上に常時保持することなく、ホストからのアクセスに応じて必要な領域のみの対応関係をこのシステム内の前記RAM上に逐次作成することを特徴とする不揮発性半導体メモリの制御方法。

【請求項 10】

前記物理ブロックを構成する不揮発性半導体メモリはフラッシュメモリであることを特徴とする請求項 8 又は 9 に記載の不揮発性半導体メモリの制御方法。

【請求項 11】

不揮発性半導体メモリシステムの制御方法において、不揮発性半導体メモリによるセルアレイを複数個の物理ブロックに分割し、

前記各物理ブロックに、このシステムが管理する論理ブロックアドレスとの対応付けの情報を記憶させ、上記物理ブロックと上記論理アドレスとの対応関係を管理するためのアドレス変換テーブルをこのシステムのRAM上に作成するに当り、

少なくとも 1 個以上の上記物理ブロックをそれぞれ集合することにより複数の領域が構成し、

所定の上記論理ブロックのアドレス範囲のデータは所定の上記領域内に格納されるように制御し、上記不揮発性半導体メモリに対するアクセス時に、該当する上記論理ブロックアドレスのデータが格納されている上記領域に対応する上記アドレス変換テーブルを、必要に応じ作成する

ことを特徴とする不揮発性半導体メモリシステムの制御方法。

【請求項 12】

不揮発性半導体メモリスラムの制御方法において、不揮発性メモリによるセルアレイを複数個の物理ブロックに分割し、前記各物理ブロックの記憶領域にこのシステムが管理する前記論理ブロックとの対応付けの情報を記憶させ、前記論理

ブロックとフラッシュメモリの物理ブロックとの対応関係を管理するためのテーブルをこのシステムのRAM上に形成する制御方法であって、1個あるいは複数個の前記物理ブロックから構成される領域を、前記フラッシュメモリのセルアレイ上に確保し、メモリアクセス時毎にアクセス対象となる前記領域について、該領域を構成する物理ブロックの記憶領域をサーチし論理ブロックと物理ブロックとの対応関係を管理する前記テーブルをこのシステムのRAM上に作成し、該テーブルを使用することにより、論理ブロックに対応する物理ブロックを選択可能としたことを特徴とする不揮発性半導体メモリシステムの制御方法。

【請求項13】

欠陥セルを含む不良物理ブロックを、選択的に冗長物理ブロックと置換する機能を有し、前記領域の各々に対して、不良物理ブロック数が所定個数以下となるよう前記置換作業を管理することを特徴とする、請求項11又は12に記載の不揮発性半導体メモリシステムの制御方法。

【請求項14】

本体に対して数種類のメモリ装置が着脱により交換可能とされた不揮発性半導体メモリシステムの制御方法において、前記本体に装着された前記メモリ装置に応じて、前記本体が有する数種類のエラー訂正手段のうちの対応するものを選択してエラー訂正を行わせることを特徴とする不揮発性半導体メモリシステムの制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、不揮発性半導体メモリシステムの制御方法に関し、特に不揮発性半導体メモリカードの制御に使用されるものである。

【0002】

【従来の技術】

近年図1に示すようなフラッシュメモリカードがデジタルスチールカメラやPDA等の携帯情報機器の記憶媒体として注目されている。フラッシュメモリとしてNAND型フラッシュメモリと呼ばれるフラッシュメモリが使用され販売がな

されている。

【0003】

このメモリカードは図1に示すように薄型のプラスチックパッケージ1にわずかな窪みが設けられておりその窪みに22ピンの平面電極を有するフラッシュメモリ2が埋め込まれている。本フラッシュメモリカードは専用のコネクタを介してホストシステムに電氣的に接続され、データの入出力を行うというものである。

【0004】

これらのメモリカードはコネクタを介して信号のやり取りを行うので、メモリカードの各端子とコネクタの端子との接触が不完全だった場合には、誤動作する可能性が大きくなる。このため、メモリカードの各端子と情報処理機器のコネクタの端子とが確実に接触しているかどうかを確認する方法が考案されてきた。

【0005】

特に、従来のメモリカードを用いた制御システムでは、メモリカードの各端子とコネクタの対応する端子は同時に接続／切断されていたので、挿入／抜去時において、接続／切断のタイミングの僅かなズレが生じることがある。このため、例えば、電源が供給されている状態で、コマンドラッチイネーブル信号等の制御信号が不定になると、意図しない書込みまたは消去コマンド等が取り込まれデータを破壊する可能性があり、メモリカードとコネクタの挿抜制御には細心の注意が必要であった。

【0006】

フラッシュメモリとして、16MビットのNAND型フラッシュメモリの場合を例にとると、図2に示すように、フラッシュメモリは512個の物理的なメモリブロックに分割されている。このブロックは消去時の最小単位となっている。1ブロックはさらに16ページに分割される。1ページは書き込みおよび読み出しの基本的な単位となる。1ページは264バイトから構成され、うち256バイトはユーザーデータ領域（データ部）、残りの8バイト（冗長部）はエラー訂正符号および管理情報等の格納（冗長部）に使用される。

【0007】

これに対して、パソコン側ではデータは図3に示す論理ブロックによって管理される。論理ブロック（LBA: Logical Block Address）は500個設定されており、1個の論理ブロックは連続した8セクタに相当する。すなわち、論理ブロック0は論理セクタ0～7を意味する。

【0008】

通常パソコン等ではデータはセクタ（512バイト）単位で管理されるため、本メモリカードでも512バイト単位でのデータ管理を基本とすべく物理ブロックの2ページをペアとして論理ブロックの1セクタ分のデータを記憶させる。データの具体的な格納方法を図4に示す。

【0009】

未使用の正常ブロックは、データ部、冗長部とも“FFh”に設定されている。下記に各々のバイトの意味あいにつき説明する。Data Area-1は512バイトデータのうち、前半の0～255バイトのデータが格納される。Data Area-2には512バイトデータのうち、後半の256～511バイトのデータが格納される。User Data Areaのデータは、ユーザに解放されており使用法はユーザに一任される。Data Status Areaはデータが正常か否かを示す。通常は“FFh”だが、正常でないデータが書き込まれている場合に“00h”が設定される。Block Status Areaはブロックが良か不良かを示す。通常は“FFh”だが、不良ブロックの場合、“00h”（初期不良ブロック）、“F0h”（後発不良ブロック）が設定される。2ビット以上“0”であった場合は、不良ブロックであると判断する。なお、本データは同一ブロック内では全て同じ値を書き込む。Block Address Area-1はブロックの論理アドレス情報を示す。なお、1論理ブロックを構成する8セクタには512の物理ブロックのうち1物理ブロックが相当するので、本データは同一ブロック内では全て同じ値が書込まれることになる。同様に、Block Address Area-2はBlock Address Area-1のデータと同じ内容が書かれている。ECC Area-1は偶数ページデータ（256バイト）の3バイトECCコードであ

る。ECC Area-2は、奇数ページデータ（256バイト）の3バイトECCコードである。

【0010】

64MビットNAND型フラッシュメモリの場合を例にとると、図5に示すように、フラッシュメモリは1024個の物理的なメモリブロックに分割されている。このブロックは消去時の最小単位となっている。1ブロックはさらに16ページに分割される。1ページは書き込みおよび読み出しの基本的な単位となる。1ページは528バイトから構成され、うち512バイトはユーザーデータ領域（データ部）、残りの16バイト（冗長部）はエラー訂正符号および管理情報等の格納（冗長部）に使用される。論理ブロックは図6に示すように、1000個に設定されており、1個の論理ブロックは連続した16セクタに相当する。すなわち論理ブロック0は論理セクタ0～15を意味する。64MビットNAND型フラッシュメモリのデータ格納方法を図7に示す。

【0011】

このようなメモリカードの制御では、データ更新時は消去済み領域に更新データを書き込み、元のデータが存在する領域を消去するという、追加書き込み方式を採用しているため、ある論理ブロックに対応するデータが存在する物理ブロックは、固定ではなく、常にメモリ内を移動している。したがって、図8に示すように物理ブロックの冗長部には自分がどの論理ブロックに対応するデータを保持しているかを示す論理ブロックアドレス情報を記憶している。ここで、図4および図7中のBlock Address Area-1及びBlock Address Area-2が、対応する論理ブロックアドレスに相当する。

【0012】

ゆえに、通常は電源投入時に、全物理ブロックの該論理ブロックアドレス情報格納領域をサーチし、システムRAM上に、論理ブロックと物理ブロックの変換テーブルを作る。一度テーブルを作成した後は、該テーブルを参照すれば、論理ブロックに対応する物理ブロックがすぐに判断可能なため、全ブロックのサーチ動作は電源投入時に1回行われる。当然のことながら、データの更新を行い、対応する物理ブロックの位置が変化した場合、メモリシステムは、論理番地／物理

番地変換テーブルの更新作業を行い、次のアクセスに備えることになる。

【0013】

しかし、従来のメモリシステムにおいては、前記論理番地／物理番地変換テーブルに要するRAM領域が膨大になってしまうという第1の問題点があった。以下にその内容を詳述する。

【0014】

従来の16MビットNAND型フラッシュメモリの論理番地／物理番地変換テーブルを図9に示す。上述したように、1論理ブロック分のデータ、すなわち連続した8セクタ分のデータは、フラッシュメモリ中の512ブロックのうちのいずれかの物理ブロックに存在する。512個の物理ブロックから1個のブロックを選択するには、9ビット必要となる。ソフトウェアの利便性に配慮してオフセットがそのまま物理ブロックを指し示す様にテーブルを構成すると、1論理ブロックに対し2バイト、計1KBのRAM領域が必要となる。例えば、論理ブロック5の情報が格納されている物理ブロックの番地はテーブルの先頭からオフセット5ワード（10バイト）目に格納される。

【0015】

このように、従来用いられていた方法においては、論理番地／物理番地変換テーブルが必要とするRAM領域が非常に大きいという問題点があった。通常よく用いられる、汎用CPUでは内蔵RAMとして1KB程度のRAMを搭載しているのが一般的である。従って、従来は、論理番地／物理番地変換テーブルだけで1KBを使用する必要があり、内蔵RAMのみではシステム構成ができず、外づけのRAMをシステムとして有することが条件となりコストアップの大きな要因となっていた。

【0016】

図10に従来の64MビットNAND型フラッシュメモリの論理番地／物理番地変換テーブルを示す。この場合、1論理ブロック分のデータ、すなわち連続した16セクタ分のデータはフラッシュメモリ中の1024ブロックのうちのいずれかの物理ブロックに存在する。1024個の物理ブロックから1個の物理ブロックを選択するには、10ビット必要となることから、合計2KBのRAM領域

が必要となる。このため、16ビットNAND型フラッシュメモリ同様膨大なRAM領域が必要となる。

【0017】

この問題は、フラッシュメモリの容量が大きくなるほど深刻になる。例えば、1Gビットの時代になるとブロック数は8192個になるため、16KBのRAM容量が必要となる。

【0018】

また、さらにメモリ容量が増加すると、フラッシュメモリの物理ブロックの冗長部に論理アドレスを格納できないという第2の問題が生じる。図7に示す16MビットNAND型フラッシュメモリの物理ブロックの冗長部のBlock Address Areaには自分がどの論理ブロックに対応するデータを保持しているかを示す論理ブロックアドレス情報が格納される。各物理ブロックの冗長部のBlock Address Areaの構造を図11に示す。図11中で、偶数ページの262バイトおよび奇数ページの259バイト目のD4～D7の4ビットと、偶数ページの263バイトおよび奇数ページの260バイト目のD0の1ビットは“1”固定値となる。従って、格納可能なブロックアドレスの最大値は、BA0～BA10で表される、2047である。512MビットのNAND型フラッシュメモリでは物理ブロックが4096個存在するため、Block Address Areaの記述方法を変更しない限り、アドレスを格納することはできない。従来のフラッシュメモリと冗長部の制御方法を異にするため、ホストが2種類のプログラムを用意しない限り、広範囲の容量のフラッシュメモリを制御できず、ホストシステムのプログラム格納領域の容量を圧迫する問題が発生する。

【0019】

次に、フラッシュメモリの書き込み、消去について簡単に述べる。フラッシュメモリの書き込みはページ単位で一括で実行される。64MビットNAND型EEPROMの場合は1ページ528バイトである。また消去はブロック単位で実行される。64MビットNAND型EEPROMは16ページで1ブロックを構成する。このようにNAND EEPROMでは書き込みと消去の単位が異なる

。従ってあるページのみを消去してデータを更新する事は出来ない。

【0020】

フラッシュメモリカードをパソコンで使用する場合には一般的にDOS支配下のドライブとして取り扱われる。図12に従来のDOSフォーマットパラメータを示す。ここでは、図12(a)として、クラスタサイズ4KBの場合、図12(b)としてクラスタサイズが8KBの場合を示している。クラスタとはDOSのファイル管理の基本最小単位で、ファイルサイズが非常に小さくても1個のクラスタ分の容量は占有する。ファイルサイズが大きい場合は、複数のクラスタのチェーンとして管理され、その管理情報はFAT (File Allocation Table)に格納される。クラスタのサイズやFATの管理方法等はブートセクタと呼ばれるセクタ内で管理される。1個のデバイスが複数のドライブとして管理される場合は、マスターブートセクタにその情報が格納される。ファイルの書き込みとしては、OSからクラスタ単位で書き込み命令が発行される。

【0021】

クラスタサイズが4KBの場合を図12(a)に示す。論理セクタ0にマスターブートセクタ、論理セクタ16にブートセクタ、論理セクタ17～22にFAT、論理セクタ23～28にFATのコピー、論理セクタ29～44にディレクトリー、論理セクタ45以降にファイルデータ領域が配置されている。

【0022】

クラスタサイズが8KBの場合を図12(b)に示す。論理セクタ0にマスターブートセクタ、論理セクタ16にブートセクタ、論理セクタ17～19にFAT、論理セクタ20～22にFATのコピー、論理セクタ23～38にディレクトリー、論理セクタ39以降にファイルデータ領域が配置されている。

【0023】

まず、図13を用いて、クラスタサイズ4KBの場合を例に従来の書き換えシーケンスを説明する。クラスタサイズは4KBなので、連続した8セクタ分の書き込み命令がOSから発行される。このとき論理セクタの45～52(クラスタA)に対する書き込み(データ更新)が発生する。

- 1) 消去済み新領域を探し、論理セクタ32～44までを元ブロックから新領

域NAND BlockCにコピーする。

- 2) 論理セクタ45～47の新データを新領域NAND BlockCに書き込む。
- 3) 元ブロックNAND BlockAを消去する。
- 4) 論理番地／物理番地変換テーブルの更新。
- 5) 消去済み新領域を探し、論理セクタ－48～52の新データを新領域NAND Block Dに書き込む。
- 6) 元ブロックNAND Block Bの論理セクタ－53～63のデータを新領域NAND Block Dにコピーする。
- 7) 元ブロックNAND BlockAを消去する。
- 8) 論理番地／物理番地変換テーブルの更新。

従って、外部からみて8セクタの書き換えを実施した場合、実際デバイスとしては、論理セクタ32～63の合計32セクタ（32ページ）の書き込み動作と、NAND BlockA、NAND Block Bの合計2ブロックに対する消去動作が実施されたことになる。

【0024】

次に、図14を用いて、クラスタBに対する書き込みシーケンスを説明する。この場合論理セクタの53～60（クラスタB）に対する書き込み（データ更新）が発生する。

- 1) 消去済み新領域を探し、論理セクタ－48～52までを元ブロックNAND Block Bから新領域NAND Block Cにコピーする。
- 2) 論理セクタ－53～60の新データを新領域NAND Block Cに書き込む。
- 3) 論理セクタ－61～63までを元ブロックNAND Block Bから新領域NAND Block Cにコピーする。
- 4) 元ブロックNAND Block Bを消去する。
- 5) 論理番地／物理番地変換テーブルの更新。

従って、外部からみて8セクタの書き換えを実施した場合、実際デバイスとしては、論理セクタの48～63の合計16セクタ（16ページ）の書き込み動作と、NAND Block Bの1ブロックに対する消去動作が実施されたことになる。

【0025】

次に、図15を用いて、クラスタサイズ8KBの場合を例に従来の書き込みシーケンスを説明する。クラスタサイズは8KBなので、連続した16セクタ分の書き込み命令がOSから発行される。このとき論理セクタの39～54（クラスタA）に対する書き込み（データ更新）が発生する。

- 1) 消去済み新領域を探し、論理セクタ32～38までを元ブロックNAND Block Aから新領域NAND Block Cにコピーする。
- 2) 論理セクタ39～47の新データを新領域NAND Block Cに書き込む。
- 3) 元ブロックNAND Block Aを消去する。
- 4) 論理番地／物理番地変換テーブルの更新。
- 5) 消去済み新領域を探し、論理セクタ48～54の新データを新領域NAND Block Dに書き込む。
- 6) 元ブロックNAND Block Bの論理セクタ55～63のデータを新領域NAND Block Dにコピーする。
- 7) 元ブロックNAND Block Bを消去する。
- 8) 論理番地／物理番地変換テーブルの更新。

従って、外部からみて16セクタの書き換えを実施した場合、実際デバイスとしては、セクタ32～63の合計32セクタ（32ページ）の書き込み動作と、NAND Block A、NAND Block Bの合計2ブロックに対する消去動作が実施されたことになる。

【0026】

上記クラスタが4KBのものと8KBの場合を同じ8KBのデータが書き込まれる場合を想定し比較すると、クラスタサイズが4KBの場合は、2回の書き込みに処理が分割され合計48セクタの書き込みおよび3ブロック分の消去動作が発生する。これに対し、クラスタサイズが8KBの場合は、1回の書き込みに処理が集約され合計32セクタの書き込みおよび2ブロック分の消去動作が発生することになる。

【0027】

このように従来のメモリシステムは、外部から見て更新されたセクタ数に比較

し、実際にデバイス上で実行された書き込み動作および消去動作がはるかに多くなるため、外部からみた書き換え速度が遅くなるという第2の問題点があった。

【0028】

次に、従来のメモリシステムにおけるファイル消去コマンド実行時の動作について説明する。通常DOSのファイルシステムにおいては、ファイル消去コマンド実行時に、ディレクトリー上に該当ファイルが無効である旨のマーキングを行い、該当ファイルが占有していたメモリ領域をFAT (File Allocation Table) 上で開放する。従って、ファイル本体のデータ部分はフラッシュメモリ上に消去されずに残っている。消去コマンド実行時の管理領域とデータ領域の関係を図16に示す。図16において、例えばFile-1とFile-4の消去コマンドを実行した場合、管理領域のFile-1とFile-4が開放されdel. markがマーキングされる。このときデータ領域においてはFile-1、File-4は消去されない。

【0029】

このため、その後の書き込みコマンド実行時に、前記開放された領域に対し新たなファイルのデータ部分が書き込まれる際に、まずフラッシュメモリの消去動作が必要となる。このため、ファイル書き込み時に必ずフラッシュメモリの消去動作を伴い、ファイル書き込み速度を劣化させるという第3の問題点が生じる。

【0030】

図4に示す、ECC Area-1は偶数ページデータ(256バイト)の3バイトECCコードである。ECC Area-2は、奇数ページデータ(256バイト)の3バイトECCコードである。

【0031】

ここでECC (Error Correction Code)とはエラー訂正のための符号をさす。システムはこのエラー訂正用の符号を利用し、読み出したデータにエラーがあるか否かを判定し、エラーが存在する場合、エラーを訂正することができる。必要なエラー訂正能力はフラッシュメモリ自身の信頼性、例えばメモリのセル構造に依存する。フラッシュメモリには複数のデータ記憶方法があり、これらのフラッ

シュメモリを例えばデジタルスチルカメラやPDA等のシステムに用いた場合のエラー訂正について考える。

【0032】

例えば、第1のフラッシュメモリカードは、図17に示すようにメモリセルのしきい値に応じた、“0”と“1”の2値の値を保持し、1ページ（256バイト）に対して1ビットエラー訂正用の符号を持たせている。また、第2のフラッシュメモリカードは、図18に示すようにメモリセルのしきい値に応じた、“00”と“01”と“10”と“11”の4値（2ビット分）の値を保持し、1個のメモリセルが破壊されると2ビット分のデータが破壊される可能性があるため、1ページ（256バイト）に対して2ビットエラー訂正用の符号を持たせている。1ビットエラー訂正用と2ビットエラー訂正用では符号の生成、およびエラーの検出、訂正に関してはアルゴリズムが異なる。

【0033】

従来のシステム（例えば、デジタルスチルカメラや、PDA等）では図19に示すように1種類のエラー訂正アルゴリズムのみ搭載していた。このため、上記第1および第2のどちらかのフラッシュメモリカードしか読めないという第4の問題点が生じ、フラッシュメモリの市場での汎用性を高める上での障害となっていた。

【0034】

【発明が解決しようとする課題】

さらに、第1、第4、第5の実施例に係る発明は、上記第1の問題点を鑑みてなされ、論理番地／物理番地変換テーブルで必要となるRAMの容量を低減し、汎用CPUの内蔵RAMのみでフラッシュメモリを制御する方法を提供し、これにより従来必要であった外付けRAMを不要となし、大幅なコストダウンを実現することを目的とする。

【0035】

さらに、本発明の第2の実施例に係る発明は、上記第2の問題点を鑑みてなされ、物理ブロックにおいて、DOS上のファイル管理の基本単位であるクラスタの区切れが、消去の単位となるブロックをまたがないようにする方法を提供し、

データの高速書込みを実現することを目的とする。

【0036】

また、本発明の第3の実施例に係る発明は、上記第3の問題点を鑑みてなされ、消去コマンド実行時に物理ブロックの管理領域を解放すると同時に、そのデータ領域の消去も行うことにより、その後の書込みコマンド実行時の処理速度の向上を実現することを目的とする。

【0037】

さらに、本発明の第6の実施例に係る発明は、上記第4の問題点を鑑みてなされ、2値/多値等の2種類のフラッシュメモリカードのいずれもまたは更に複数のフラッシュメモリがサポート可能なシステムを提供することにある。

【0038】

【課題を解決するための手段】

上記目的を達成する手段として、本発明の第1の実施例に係る発明では、システムが管理する論理ブロックと、複数のメモリセルにより構成され、前記論理ブロックに対応するデータを記憶する物理ブロックと、前記物理ブロック中に含まれ、前記対応する論理ブロックのアドレスを記憶する冗長部と、前記物理ブロックが少なくとも2以上で構成される物理ブロックエリアとを具備し、前記論理ブロックと前記物理ブロックエリアとの対応関係を管理するための論理番地/物理番地変換テーブルを作成するメモリシステムの制御方法を提供する。

【0039】

また、上記メモリシステムの制御方法において、メモリアクセス時に、前記論理番地/物理番地変換テーブルを参照し前記論理ブロックに対応する物理ブロックエリアのアドレスを読み出し、前記物理ブロックエリアを構成する少なくとも2以上の物理ブロックの前記冗長部に記憶される対応する論理ブロックのアドレスを読み出すことにより、前記論理ブロックと対応する物理ブロックを選択するメモリシステムの制御方法を提供する。

【0040】

さらに、前記論理番地/物理番地変換テーブルを電源投入時に、作成するメモリシステムの制御方法を提供する。

【0041】

また、上記目的を達成する手段として、本発明の第2の実施例に係る発明では、データの高速書き換えを実現する手段として、本発明ではシステムが管理するファイルと、少なくとも1以上の前記ファイルを記憶する複数のブロックとを有し、前記ブロック内で、前記ファイルの先頭が前記ブロックの先頭部と一致するように配置されることを特徴とするメモリシステムの制御方法を提供する。

【0042】

さらに、前記ブロックは消去時の最小単位であることを特徴とするメモリシステムの制御方法を提供する。

【0043】

また、上記目的を達成する手段として、本発明の第3の実施例に係る発明では、システムが管理するファイルと、前記ファイルの内容を記憶するデータ領域と、前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークし、対応する前記データ領域の消去を行うことを特徴とするメモリシステムの制御方法を提供する。また、消去エリアへのデータの高速書込みを実現する第2の手段として、本発明ではシステムが管理するファイルと、前記ファイルの内容を記憶するデータ領域と、前記ファイルとデータ領域の対応関係を記憶する管理領域とを有し、前記ファイルの消去を行う時に、前記管理領域に対応するデータ領域が空き領域である旨をマークしておき、メモリシステムに入力される信号に基づいて、前記管理領域の内容を検知し対応するデータ領域の消去を行うことを特徴とするメモリシステムの制御方法を提供する。

【0044】

さらに、上記目的を達成する手段として、本発明の第4の実施例に係る発明では、不揮発性半導体メモリシステムの制御方法において、セルアレイを複数の物理ブロックに分割し、前記各物理ブロックにこのシステムが管理する前記論理ブロックとの対応付けの情報を記憶させ、前記論理ブロックと前記物理ブロックとの対応関係を管理するためのテーブルを前記システムのRAM上に保持するに当り、前記論理ブロックと前記物理ブロックの対応関係のうち、ホストからのア

クセスに応じて必要な領域のみの対応関係を前記システム内の前記RAM上に逐次作成することを特徴とする不揮発性半導体メモリシステムの制御方法を提供する。

【0045】

さらに、上記目的を達成する手段として、本発明の第5の実施例に係る発明では、不揮発性半導体メモリシステムの制御方法において、不揮発性半導体メモリによるセルアレイを複数個の物理ブロックに分割し、前記各物理ブロックに、このシステムが管理する論理ブロックアドレスとの対応付けの情報を記憶させ、上記物理ブロックと上記論理アドレスとの対応関係を管理するためのアドレス変換テーブルをこのシステムのRAM上に作成するに当り、少なくとも1個以上の上記物理ブロックをそれぞれ集合することにより複数の領域が構成し、所定の上記論理ブロックのアドレス範囲のデータは所定の上記領域内に格納されるように制御し、上記不揮発性半導体メモリに対するアクセス時に、該当する上記論理ブロックアドレスのデータが格納されている上記領域に対応する上記アドレス変換テーブルを、必要に応じ作成することを特徴とする不揮発性半導体メモリシステムの制御方法を提供する。また、不揮発性半導体メモリシステムの制御方法において、不揮発性メモリによるセルアレイを複数個の物理ブロックに分割し、前記各物理ブロックの記憶領域にこのシステムが管理する前記論理ブロックとの対応付けの情報を記憶させ、前記論理ブロックとフラッシュメモリの物理ブロックとの対応関係を管理するためのテーブルをこのシステムのRAM上に形成する制御方法であって、1個あるいは複数個の前記物理ブロックから構成される領域を、前記フラッシュメモリのセルアレイ上に確保し、メモリアクセス時毎にアクセス対象となる前記領域について、該領域を構成する物理ブロックの記憶領域をサーチし論理ブロックと物理ブロックとの対応関係を管理する前記テーブルをこのシステムのRAM上に作成し、該テーブルを使用することにより、論理ブロックに対応する物理ブロックを選択可能としたことを特徴とする不揮発性半導体メモリシステムの制御方法を提供する。

【0046】

さらに、上記目的を達成する手段として、本発明の第6の実施例に係る発明で

は、本体に対して数種類のメモリ装置が着脱により交換可能とされた不揮発性半導体メモリシステムの制御方法において、前記本体に装着された前記メモリ装置に応じて、前記本体が有する数種類のエラー訂正手段のうちの対応するものを選択してエラー訂正を行わせることを特徴とする不揮発性半導体メモリシステムの制御方法を提供する。

【0047】

【発明の実施の形態】

本発明の第1の実施例について詳細に説明する。

本実施例は、論理番地／物理番地変換テーブルに必要なRAM領域の容量の削減を行ったフラッシュメモリカードについてである。

【0048】

まず、繰返しになるが本発明のメモリシステムの、物理ブロックの構造と論理ブロックの関係について説明する。

【0049】

本発明のメモリシステム物理ブロック構造及び物理ブロック内のデータ構成は、従来技術の項で説明したもの同様に、図2および図4に示される。以下に、図4を用いて、本発明のメモリシステムにおける物理ブロック内の各々のバイトの意味を説明する。

【0050】

Data Area-1は512バイトデータのうち、前半の0～255バイトでデータが格納される。Data Area-2には512バイトデータのうち、後半の256～511バイトのデータが格納される。User Data Areaのデータは、ユーザに解放されており使用方法はユーザに一任される。Data Status Areaはデータが正常か否かを示す。通常は“FFh”だが、正常でないデータが書き込まれている場合に“00h”が設定される。Block Status Areaはブロックが良か不良化を示す。通常は“FFh”だが、不良ブロックの場合、“00h”（初期不良ブロック）、“F0h”（後発不良ブロック）が設定される。2ビット以上“0”であった場合は、不良ブロックであると判断する。なお、本データは同一ブロック内では全て同

じ値を書き込む。Block Address Area-1はブロックの論理アドレス情報を示す。なお、1論理ブロックを構成する8セクタには512の物理ブロックのうち1物理ブロックが相当するので、本データは同一ブロック内では全て同じ値が書き込まれることになる。同様に、Block Address Area-2はBlock Address Area-1のデータと同じ内容が書かれている。ECC Area-1は、偶数ページデータ(256バイト)の3バイトECCコードである。ECC Area-2は、奇数ページデータ(256バイト)の3バイトECCコードである。

【0051】

ここで、ECCについて説明する。本実施例では256バイト(2048ビット)単位のデータに対しECC符号を生成する。1ビット訂正の機能を持たせるために、256バイトに対し22ビットのECCデータを使用している。256バイトは図20に示すように、データに並べたものである。

【0052】

即ち、1Byte目の入力のbit 0が2048bitの1bit目(アドレス; 00000000 000)となり、

256Byte目の入力のbit 7が2048bitの2048bit目(アドレス; 11111111 111)となる。

【0053】

ECCコード(ラインパリティ(LP)とカラムパリティ(CP))は図21に示す条件を満足する1024ビットの奇数パリティとして算出される。

【0054】

カラムパリティ-CP0~CP5は1バイト(8ビット)でデータが入力される毎に更新される。ECC符号の生成をソフトウェアで行う場合は、1バイトの入力(256通り)に対してのカラムパリティ計算結果を予めシステム内のROM上に持つ方法が考えられる。この方法により、ビット単位の演算が不要になり、計算時間が大幅に短縮できる。またバイトの入力(256通り)に対してのカラムパリティ計算結果を電源投入時に一括で計算し、RAM上に保持する方法も考えられる。前案に比較しROMは無くても済むが、代わりにRAM領域が必要と

なる。

【0055】

図22は、本発明の第2の実施例に係るフラッシュメモリカードの電源投入時の制御フローチャートである。以下にフローに従い説明する。

(ステップS1) コネクタにメモリカードが挿入されたことを受けて、電源電圧の検知を行う。

(ステップS2) メモリカードのIDコードを読み出し、記憶容量を読みとる。

(ステップS3) もしシステムがサポートしていないIDコードが読み出されたらリジェクトする。

(ステップS4) 物理フォーマットの確認を行う。物理番地の先頭ブロックの情報を読む。

(ステップS5) もしシステムがサポートしていないフォーマットがなされていればリジェクトする。

(ステップS6) 論理番地／物理番地変換テーブルを作るとともに、次の書き込み動作で書き込みを行うための消去済み領域を選択する。

【0056】

図23に上記ステップS6に示した電源投入時に作成する論理番地／物理番地変換テーブルの作成のフローチャートを示す。

(ステップS1) 論理番地／物理番地変換テーブルが入るRAM領域をリセットする。

(ステップS2) 次のデータ書き込みに使用される消去済みのブロックを記憶するテーブル領域をリセットする。

(ステップS3) 物理ブロックの1からサーチを開始する。

(ステップS4) ブロックの冗長部を読み出す。

(ステップS5) 所定領域のデータを元に該ブロックが正常なブロックか否かを判断する。不良ブロックの場合以下に続く処理は不要となり、次のブロックのサーチに移行する。

(ステップS6) 消去済みのエリアかどうかを判断する。

(ステップ S 7) 消去済みのブロックであれば次の書き込みの際に使用するブロックの候補としてテーブル上に格納する。

(ステップ S 8) 消去済みのエリアで無ければ、論理アドレス情報領域を抽出する。このとき、パリティチェックを行い妥当性を確認する。

(ステップ S 9) 上記内容をもとに論理番地／物理番地変換テーブルを作成する。

(ステップ S 10) 物理ブロック番号のカウントアップ。

(ステップ S 11) 512 ブロックサーチしたら終了。

【0057】

このフローにしたがって作成された論理番地／物理番地変換テーブルを図 24 に示す。図 24 に示す物理ブロックエリア (Physical Block Area) とは、連続した 2 個の物理ブロックの集合体を意味する。例えば、物理ブロックエリア 0 とは、物理ブロック 0 と物理ブロック 1 を示す。本テーブルでは 1 個の論理ブロックに対して 1 個の物理ブロックエリアを割り付けている。例えば、論理ブロック 0 に対し、物理ブロックエリア 5 が割り振られている場合、物理ブロックの 10 もしくは 11 が実際の論理ブロック 0 のデータを記憶していることになる。従って、実際にアクセスする時には、物理ブロック 10 および 11 の冗長部の論理アドレスとの関連を示すデータ領域を検索し、どちらが論理ブロック 0 のデータを本当に格納しているのかを判断する必要がある。しかし、極めて限定された領域をリードするのみで済むため、メモリアクセスの性能に及ぼす影響はほとんどない。

【0058】

このとき、物理ブロックエリアは全体で 256 個 (512 / 2) 存在し、8 ビットのデータによって記述する事が可能となる。ソフトウェアの利便性に配慮してオフセットがそのまま物理ブロックを指し示す様にテーブルを構成すると、1 ブロックに対して 1 バイト、計 0.5 KB の RAM 領域が必要となる。例えば、論理ブロック 5 の情報が格納されている物理ブロックエリアの番地はテーブルの先頭からオフセット 5 バイト目に格納される。

【0059】

この0.5KBのRAM容量は、従来必要であった1KBのRAM容量に対し半分となる。汎用CPUは通常1KB程度のRAM領域をもっているが、本実施例によって得られた0.5KBのRAM領域の削減は非常に大きな役割をしめる。

【0060】

すなわちコストアップを招く外付けRAMを設けなくても、本実施例によって得られた0.5KBの空き領域を使用する事によってシステムを構成する事が可能となりコストダウンをはかることができる。

【0061】

また、上記実施形態に限らず、物理ブロックエリアとして4物理ブロックを定義しても良いし、さらに大きなブロック数を想定しても構わない。

【0062】

図25は本実施例に係るフラッシュメモリカードの読み出し時の動作フローチャートである。以下にフローに従い説明する。

(ステップS1) 読み出しを行う先頭セクタアドレスと転送セクタ数をホストから受け取る。

(ステップS2) 読み出し範囲が、妥当な範囲か検証する。

(ステップS3) セクタを論理ブロックに変換。16Mビット品の場合は1ブロック8セクタ構成なので8で割ることになる。

(ステップS4) 論理番地/物理番地変換テーブルを参照し、該当論理ブロックが存在する物理ブロックエリアを得る。

(ステップS5) ブロックエリアにある2個の物理ブロックの論理アドレス情報領域を調べ、いずれがホストの指定した論理ブロックのデータを格納しているかを調べる。

(ステップS6) 特定された物理ブロックから1セクタ分データを読み出す。例えばセクター番号が0の時は、物理ブロックの先頭2ページのデータを読み、例えばセクター番号が7の時は、物理ブロックの最終2ページのデータを読む。1個の物理ブロックの中では、8個のセクタのデータが順番に並んでいる。

(ステップ S 7) 読み出したデータに対しエラーチェックを行い、エラーが無いか確認する。

(ステップ S 8) エラーが検出されたら、訂正可能か否かを判断する。

(ステップ S 9) エラーが検出され、なおかつ訂正可能な場合データを訂正する。

(ステップ S 10) ホストが要求したセクタ数を読み出したら終了する。

(ステップ S 11) 次の読み出しセクターが物理ブロックの境界を越えるかどうかを判断する。例えば、セクター 7 から 8 に移行する場合、データは各々異なる物理ブロックに存在するため、新たに論理番地／物理番地変換テーブルを再度参照する。

(ステップ S 12) 同一ブロック内で読み出しを継続する場合は、読み出すページのカウンタアップする。

(ステップ S 13) 別のブロックへ移動する場合は、論理ブロックをカウンタアップし、ページのカウンタもリセットする。

【0063】

次に、本実施例の書き込み時の動作について説明する。

書き込みは基本的に次の 3 部分の処理に大別される。例えば、論理セクタ 3 を書き換える場合例に説明する。更新はセクタ 3 のみだが、セクタ 0 から 7 の 8 セクタ分のデータは同一ブロック上に存在するので、1 ブロックに対する処理が必要になる。

【0064】

第 1 に、論理セクタ 0、1 及び 2 に関しては、データの更新は無なので、論理セクタ 0、1 及び 2 のデータは、元々格納されていた論理ブロックから新しく書き込みを行う物理ブロックへコピーされる。

【0065】

第 2 に、論理セクタ 3 は更新するので、元のデータをコピーする必要はなく、ホストから与えられたデータを新たに書き込みを行うブロックへ書き込む。

【0066】

第 3 に、論理セクタ 4 ～ 7 はデータの更新はないので、論理セクタ 4 ～ 7 のデ

ータは、元々格納されていた物理ブロックから新しく書き込みを行う物理ブロックへコピーする。

【0067】

以上のように、1ブロックに対して、コピー／更新データ書き込み／コピーの動作が基本となる。勿論、書き込みがセクタ0～7の様な場合は、1ブロック分全てのデータが更新されるのでコピー動作は不要となることは自明である。以下に示すフローチャートの分岐は主にこれから書き込むセクタが更新データか、もしくはコピー動作なのかを判断しながら進行することになる。

【0068】

図26は本実施例に係るフラッシュメモリカードの書き込み時の動作フローチャートである。以下にフローに従い説明する。

(ステップS1) ホストからデータの更新を行う先頭セクタ番地と転送セクタ数を受け取る。

(ステップS2) 論理ブロック番号に変換し、論理番地／物理番地変換テーブルを参照する。読み出し動作と同様に論理ブロックエリアの2個のブロックから本当の物理ブロックを選択する。ここで選択したブロックからコピーすべきデータを吸い上げることになる。

(ステップS3) 物理ブロックの先頭から処理を開始する。

(ステップS4) ブロック前半のコピー動作かもしくはデータの更新かを判断する。

(ステップS5) コピーであれば、元ブロックからデータを読み出し、新ブロックに書き込みを実行する。

(ステップS6) 次のセクタの処理に移行する。

(ステップS7) (ステップS4) で更新領域と判断されたら、ホストから受け取った更新データを基に書き込みを行う。

(ステップS8) 次のセクタの処理に移行する。

(ステップS9) ホストが要求するセクタ数書き込んだか確認する。

(ステップS10) (ステップS9) で要求数の書き込みが終了したと判断された場合、ブロックの境界か否かを判断する。未書き込みの領域が残っていれば

、ブロック後半のコピー動作に移行。ブロック境界であれば、これ以上コピー動作を実行する必要はない。

(ステップ S 1 1) 元のブロックからデータを読み出し、新ブロックに書き込む。

(ステップ S 1 2) 次のセクタの処理に移行する。

(ステップ S 1 3) (ステップ S 4) でホストの要求するセクタ数の書き込みが終了していない場合は更なる書き込みが必要だが、ブロック境界であれば、次の物理ブロックに対する処理に移行する。

(ステップ S 1 4) 次にブロック処理に移行または、処理を終了する前に、書き込みを行った結果をもとに論理番地／物理番地変換テーブルの更新を行い、さらに元データが存在した物理ブロックを消去し、次の処理での新たな書き込み領域としての候補領域として登録する。

(ステップ S 1 5) 次のブロックの処理に移行する。

【0069】

以上に述べたように、本実施例のメモリシステムの制御方法によれば、RAM領域の大幅な削減が実現できる。これは、従来のメモリカードシステムは論理番地／物理番地変換テーブルで必要なRAM容量が大きく、汎用CPUの内蔵RAMのみではシステム構成ができず、外づけのRAMを設けることが条件となっていたに対して、本発明のメモリカードシステムは論理番地／物理番地変換テーブルとして必要なRAMの容量を低減し、汎用CPUの内蔵RAMのみ制御可能としたフラッシュメモリを制御する方法を用いるからである。このことにより、従来必要であった外付けRAMを不要とし、大幅なコストダウンを実現することが可能となる。

【0070】

次に、本発明の第2の実施例について詳細に説明する。

本実施例はDOSフォーマット形式で用いた場合に、データの書き換え動作を高速化したフラッシュメモリカードについてである。

【0071】

図27に本実施例によるDOSフォーマットパラメータを示す。ここでは、図

27 (a) として、クラスタサイズ 4 KB の場合、図 27 (b) としてクラスタサイズが 8 KB の場合を示している。クラスタサイズが 4 KB の場合、論理セクタ 0 にマスターブートセクタ、論理セクタ 19 にブートセクタ、論理セクタ 20 ~ 25 に FAT、論理セクタ 26 ~ 31 に FAT のコピー、論理セクタ 32 ~ 47 にディレクトリー、論理セクタ 48 以降にファイルデータ領域が配置されている。クラスタサイズが 8 KB の場合、論理セクタ 0 にマスターブートセクタ、論理セクタ 25 にブートセクタ、論理セクタ 26 ~ 28 に FAT、論理セクタ 29 ~ 31 に FAT のコピー、論理セクタ 32 ~ 47 にディレクトリー、論理セクタ 48 以降にファイルデータ領域が配置されている。このように、クラスタサイズが 4 KB および 8 KB の場合とも、クラスタの区切れが物理的なブロックの区切れをまたがないようパラメータが設定されている。これは、DOS フォーマットパラメータの内、ブートセクタの配置される場所を調整することにより実現できる。

【0072】

まず、図 28 を用いて、クラスタサイズ 4 KB の場合を例に書き込みシーケンスを説明する。クラスタサイズは 4 KB なので、連続した 8 セクタ分の書き込み命令が OS から発行される。このとき論理セクタの 48 ~ 55 (クラスタ A) に対する書き込み (データ更新) が発生する。

- 1) 消去済み新領域を探し、論理セクタ 48 ~ 55 までの新データを新領域 NAND Block C に書き込む
- 2) 論理セクタ 56 ~ 63 の元データを新領域 NAND Block C にコピーする。
- 3) 元ブロック NAND Block B を消去する。
- 4) 論理番地 / 物理番地変換テーブルの更新。

【0073】

従って、外部からみて 8 セクタの書き換えを実施した場合、実際デバイスとしては、論理セクタ 48 ~ 63 の合計 16 セクタ (16 ページ) の書き込み動作と、NAND Block B の 1 ブロックに対する消去動作が実施されたことになる。

【0074】

次に、図 29 を用いて、またクラスタ B に対する書き込みシーケンスを説明す

る。この場合論理セクタの56～63（クラスタB）に対する書き込み（データ更新）が発生する。

- 1) 消去済み新領域を探し、論理セクタ48～55までの元データを新領域NAND BlockCにコピーする。
- 2) 論理セクタ56～63の新データを新領域NAND BlockCに書き込む。
- 3) 元ブロックNAND BlockBを消去する。
- 4) 論理番地／物理番地変換テーブルの更新。

【0075】

従って、外部からみて8セクタの書き換えを実施した場合、実際デバイスとしては、論理セクタ48～63の合計16セクタ（16ページ）の書き込み動作と、NAND BlockBの1ブロックに対する消去動作が実施されたことになる。

【0076】

次に、図30を用いて、クラスタサイズ8KBの場合を例に書き込みシーケンスを説明する。クラスタサイズは8KBなので、連続した16セクタ分の書き込み命令がOSから発行される。このとき論理セクタの48～63（クラスタA）に対する書き込み（データ更新）が発生する。

- 1) 消去済み新領域を探し、論理セクタ48～63までの新データを新領域NAND BlockCに書き込む。
- 2) 元ブロックNAND BlockBを消去する。
- 3) 論理番地／物理番地変換テーブルの更新。

【0077】

従って、外部からみて16セクタの書き換えを実施した場合、実際デバイスとしては、論理セクタ48～63の合計16セクタ（16ページ）の書き込み動作と、NAND BlockBの1ブロックに対する消去動作が実施されたことになる。

【0078】

上記クラスタが4KBのものと8KBの場合を同じ8KBのデータが書き込まれる場合を想定し比較すると、クラスタサイズが4KBの場合は、2回の書き込みに処理が分割され合計32セクタの書き込みおよび2ブロック分の消去動作が発生する。これに対し、クラスタサイズが8KBの場合は、1回の書き込みに処

理が集約され合計 16 セクタの書き込みおよび 1 ブロック分の消去動作が発生することになる。

【0079】

図 12～図 15 に示した従来のメモリシステムの書き換え速度と比較すると、クラスタが 4 KB の場合では、8 KB のデータを更新するのに 48 セクタの書き込みおよび 3 ブロック分の消去動作が発生していたものが、32 セクタの書き込みおよび 2 ブロック分の消去動作となり、時間的に $2/3$ に短縮されたことになる。また、クラスタが 8 KB の場合では、8 KB のデータを更新するのに 32 セクタの書き込みおよび 2 ブロック分の消去動作が発生していたものが、16 セクタの書き込みおよび 1 ブロック分の消去動作となり、時間的に $1/2$ に短縮されたことになる。

【0080】

このように、DOS のファイル管理の単位であるクラスタの区切れが、フラッシュメモリの物理的なブロックの境界をまたがないことにより、書き換え速度の高速化がはかれる。

【0081】

また、クラスタのサイズに着目するとクラスタが 4 KB の場合では、8 KB のデータを更新するのに 32 セクタの書き込みおよび 2 ブロック分の消去動作が発生していたものが、クラスタが 8 KB の場合では 16 セクタの書き込みおよび 1 ブロック分の消去動作となり、時間的に $1/2$ になる。すなわち、クラスタのサイズをフラッシュメモリの物理ブロックのサイズと同じにすることによってより高速な書き込みが可能となる。勿論、クラスタのサイズがフラッシュメモリの物理ブロックのサイズの整数倍の場合も同様の効果を得ることができる。

【0082】

次に、本発明の第 3 の実施例について詳細に説明する。

本実施例は消去後の書き込みコマンド実行時の処理速度を向上させたのフラッシュメモリカードについてである。

【0083】

本実施例のフラッシュメモリカードシステムは、通常 DOS のファイルシステ

ムにおけるファイル消去と異なり、ディレクトリ上に該当ファイルが無効である旨のマーキングを行い、該当ファイルが占有していたメモリ領域を F A T (File Allocation Table) 上で開放するのみならず、ファイル本体のデータ部分をフラッシュメモリ上で消去することを特徴とする。すなわちファイルの消去命令時に、開放されたクラスタの領域に対する消去動作を実施する。

【0084】

図31に消去コマンド実行時の管理領域とデータ領域の関係を示す。図31において、例えば F i l e - 1 と F i l e - 4 の消去コマンドを実行した場合、管理領域の F i l e - 1 と F i l e - 4 が開放され d e l . m a r k がマーキングされ、さらにデータ領域において F i l e - 1 と F i l e - 4 が記憶されていた領域は消去される。

【0085】

従って、次に新たなファイル書き込み命令が発生したときに選択されるクラスタは既に消去済みであるので、即座に書き込みが可能となり、ファイル書き込み速度が向上する。一般にフラッシュメモリにおいては書き込みより消去の方が時間がかかるため、本実施例によって達成されるファイル書き込み速度の向上効果は顕著である。

【0086】

また本実施例における効果が最も顕著に現れる条件は上述した第2の実施例から分かる様に、クラスタサイズとフラッシュメモリのブロックサイズを一致させた場合である。クラスタサイズがフラッシュメモリのブロックサイズより小さい場合は、ブロックの一部分を消去することになるが、処理が煩雑であることに加え、仕様上ブロックの一部分のみが消去されている状態を許さない場合もある。クラスタサイズとブロックサイズが一致していれば、単純にブロックを消去する事によってクラスタを開放できる。勿論クラスタサイズが物理ブロックサイズの整数倍の場合も同様の効果が得られる。

【0087】

また、本実施例はその主旨を変えない範囲で様々拡張が可能である。例えば、本実施例では、ファイル消去時にデータ領域の該当するクラスタへの消去動作を

実行したが、消去動作の実行タイミングはそれに限られない。例えば、フォーマット動作を実行する時点で全クラスタの消去動作を実行しても良い。勿論メモ리카ードの出荷時点でデータ領域のクラスタを消去済みの状態で出荷すると良い。通常メモ리카ードの出荷時にはメモ리카ードの出荷試験を実施する。このテスト終了時にディレクトリーおよびFATの書き換えによって、ファイルが無い状態にするだけでなく、本実施例のようにデータ領域の消去動作を実行しておく、エンドユーザーの手に渡った時点で、ユーザーが手を加えることなく高速なファイル書き込みが期待できる。

【0088】

さらに例えばデジタルスチルカメラ等で使用する場合を考えると、カメラ上でのファイルの消去、再フォーマット等の画像ファイルの消去を伴う動作時において、FAT等の書換のみならず、ファイル本体が格納されていたクラスタ領域の消去を同時に実行すれば、その後の画像書き込みシーケンスにおいて、高速書き込みが可能となり、連写や動画の取り込み等が可能となる。この際には、クラスタサイズがデバイスのブロックサイズの正数倍で、クラスタの区切れがブロックサイズの区切れと一致しているとファイル本体部分の消去が容易となる。またデジタルカメラ等で電源の投入時等に自動的に不要なファイル部分を消去するようにしても良い。こうすれば、パソコン上で単純にFAT等の更新によってファイル消去がなされていたメモ리카ードに対しても、ユーザーの手を煩わせることなく書き込み高速化が計れる。またそのタイミングは電源投入時にのみならずいつでもかまわない。また、本発明はコンピュータシステム上で本メモ리카ードを使用する場合も同様の手段により書き込みの高速化が計れる。

【0089】

また、本実施例によれば、データの書換時に発生する、消去および書き込み回数を減らすことが可能なため、一般的に書換回数に制限のあるフラッシュメモリを使用する場合には、メモリの寿命をのばす効果も有する。

【0090】

以上に述べたように、本実施例のメモリスシステムの制御方法によれば、DOSのファイル管理の単位であるクラスタの区切れが、フラッシュメモリの物理的な

ブロックの境界をまたがないことにより、データの書換時に発生する消去および書き込み回数を減らすことが可能なため、書き換え速度の高速化が可能となる。さらに、一般的に書換回数に制限のあるフラッシュメモリを使用する場合には、メモリの寿命をのばすことも可能となる。

【0091】

次に、本発明の第4の実施例について詳細に説明する。

本実施例は、論理番地／物理番地変換テーブルに必要なRAM領域の容量の削減を行ったフラッシュメモリカードについてである。

【0092】

本実施例では、論理番地／物理番地変換テーブル作成に際し、ホストからのアクセスを2種類に分類する。例えば、ケース1は論理ブロック500個の内、前半の250個の論理ブロックをアクセスしている状態、ケース2は論理ブロック500個の内、後半の250個の論理ブロックをアクセスしている状態とする。ケース1の場合は、論理番地／物理番地変換テーブルには前半250ブロック分のテーブルを保持させ、上記ケース2の場合は、論理番地／物理番地変換テーブルは後半250ブロック分のテーブルを保持させる。ケース1の場合のテーブルの状態を図32(a)に示し、ケース2の場合のテーブルの状態を図32(b)に示す。

【0093】

ある瞬間、テーブル上に論理ブロックの前半250個分のテーブルが存在していたとすると、ホストからのアクセス範囲が論理ブロックの後半0～249ならば、既存のテーブルを使用して論理ブロックー物理ブロックの対応を検索することができる。

【0094】

同様にある瞬間、テーブル上に論理ブロックの後半250個分のテーブルが存在していたとすると、ホストからのアクセス範囲が論理ブロックの後半250～499ならば、既存のテーブルを使用して論理ブロックー物理ブロックの対応を検索することができる。

【0095】

次に、ある瞬間、テーブル上に論理ブロックの前半250個分のテーブルが存在していたとし、ホストからのアクセス範囲が論理ブロックの後半250～499ならば、既存のテーブルを使用して論理ブロックー物理ブロックの対応を検索することができない。従ってこの場合は、論理ブロックの後半250個分に相当する論理番地／物理番地変換テーブルを作り直す。これにはフラッシュメモリの全領域の再度の参照が必要となる。

【0096】

同様に、ある瞬間、テーブル上に論理ブロックの後半250個分のテーブルが存在していたとし、ホストからのアクセス範囲が論理ブロックの前半0～249ならば、既存のテーブルを使用して論理ブロックー物理ブロックの対応を検索することができない。従ってこの場合は、論理ブロックの前半250個分に相当する論理番地／物理番地変換テーブルを作り直す。これにはフラッシュメモリの再度の参照が必要となる。

【0097】

このように、ホストからアクセスした領域に相当する論理番地／物理番地変換テーブルがRAM上になれば再度フラッシュメモリの全領域を参照し必要な論理番地／物理番地変換テーブルを作りなおす。

【0098】

この場合の論理番地／物理番地変換テーブル作成のフローチャートを図33に示す。

(ステップS0) アクセスの最初で既に必要な論理番地／物理番地変換テーブルの有無を確認し必要であればテーブル作成ルーチンへ移行。

(ステップS1) 論理番地／物理番地変換テーブルが入るRAM領域をリセットする。

(ステップS2) 物理ブロックの先頭からサーチを開始する。

(ステップS3) ブロックの冗長部を読み出す。

(ステップS4) 所定領域のデータを元に該ブロックが正常なブロックか否かを判断する。

【0099】

不良ブロックの場合以下に続く処理は不要となり、次のブロックのサーチに移行する。

(ステップS5) 消去済みのエリアかどうかを判断する。

(ステップS6) 消去済みのブロックであれば次の書き込みの際に使用するブロックの候補としてテーブル上に格納する。

(ステップS7) 消去済みのエリアで無ければ、論理アドレス情報領域を抽出する。

(ステップS8) 上記内容をもとに論理番地／物理番地変換テーブルを作成する。

(ステップS9) 物理ブロック番号のカウントアップ。

全ブロックサーチしたら終了。

【0100】

以上のような動作で必要に応じ論理番地／物理番地変換テーブルを作成する。

【0101】

また本発明は上記実施例にかぎられない。主旨を逸脱しない範囲で種々変更が可能である。

【0102】

例えば本実施例では、フラッシュメモリを前半、後半の2分割したが、分割の数は2個に限定されず、任意の数に分割できる。

【0103】

また本実施例ではフラッシュメモリを前半、後半の等しい大きさに分割したが、本発明はこれに限られない。このように分割の個数、また大きさは自由度を有する。

【0104】

3個以上の領域に分割した際、常に1個のみのテーブルを保持する必要はない。たとえば本フラッシュメモリカードがDOSのデバイスとして、画像ファイル等を格納する場合を考えてみる。通常デバイスの先頭には、ファイルの管理領域

、すなわちマスターブートセクタ、パーティションブートセクタ、ファイル・アロケーション・テーブル（FAT）、ディレクトリ領域が存在する。これらのファイル管理領域はファイルの更新が行われたり、アクセスされるたび頻繁にアクセスされる領域である。この場合このファイル管理領域に相当するエリアを1個の領域、その他のファイルデータ格納領域を2分割する。ファイル管理領域に相当する領域の論理番地／物理番地変換テーブルは、常に保持するようにしても良い。この場合、論理アドレスの後半部分にファイルを書き込む動作を行う際、ファイル管理領域と論理アドレス後半のテーブルを交互に頻繁に作り直す必要はなくなるので性能の劣化が防げる。ここでファイルの管理領域と記述したが、ファイル管理領域以上の領域に対しテーブルを常に保持すれば同様の効果が得られる。

【0105】

以上に述べたように本実施例によれば、論理ブロックと物理ブロックの対応関係のすべてをRAM上に常時保持するのではなく、ホストからのアクセスに応じて必要な領域のみの対応関係をシステム内のRAM上に逐次作成するので、全領域の対応関係を常時RAM上に保持する場合に比べ、システムとして最低限必要なRAM領域を低減でき、これまで不可能であった汎用CPUの内蔵RAMのみでメモリを制御することが可能となり、外付けRAMを有していた従来のケースに対して大幅なコスト削減が可能となる。

【0106】

次に、本発明の第5の実施例について詳細に説明する。

本実施例は、論理番地／物理番地変換テーブルに必要なRAM領域の容量の削減を行ったフラッシュメモリカードについてである。

【0107】

本実施例では、フラッシュメモリの物理ブロックアドレスを複数の論理的な領域（以下ゾーンと呼ぶ）に分割する。このとき各ゾーンに割り当てられる物理ブロック数、すなわちゾーン毎の容量は均一であっても、不均一であってもかまわない。またゾーンの数も1個であっても複数個であってもかまわない。また偶数個であっても奇数個であってもかまわない。

【0108】

図34は、16MビットNAND型フラッシュメモリを2つの均等な容量のゾーンに分割した場合の物理ブロックの構造と論理アドレスと物理アドレスの関係を示した図である。

【0109】

ホストが該フラッシュメモリを制御する場合に、必要となる論理ブロック数は500個と定義され、論理ブロックアドレス0～499の値をとる。物理ブロックの冗長部には自分がどの論理ブロックに対応するデータを保持しているかを示す論理ブロックアドレス情報が格納されるが、本実施例では、論理ブロックアドレスは、ゾーン毎に0からシリーズに付与される。従って、ゾーン1には論理ブロックアドレス0～249が割り当てられ、それぞれの論理ブロックは物理ブロックアドレス0～255の256個の物理ブロックのいずれかに対応する。また物理ブロックの冗長部に格納される論理ブロックアドレスは0～249のいずれかが格納される。また、ゾーン2には論理ブロックアドレス250～499が割り当てられ、それぞれの論理ブロックは物理ブロックアドレス256～511の256個の物理ブロックのいずれかに対応する。また物理ブロックの冗長部に格納される論理ブロックアドレスは0～249のいずれかが格納される。

【0110】

データ更新時の追加書き込み方式は、更新対象となる論理ブロックアドレスを含むゾーン内で行われ、ある論理ブロックに対応するデータが存在する物理ブロックは、固定ではなく、常に各ゾーン内で移動している。

【0111】

本実施例では、通常は電源投入時に、ゾーン内の全物理ブロックの冗長部に格納された論理ブロックアドレス情報をサーチし、システムRAM上に論理ブロックと物理ブロックの変換テーブルを作成するが、RAM上に作成するゾーンは、どのゾーンについて行ってもかまわない。通常、該フラッシュメモリがDOS管理下で使用される場合、管理情報であるFATや、ディレクトリ領域等が最初のゾーンに置かれるため、先頭のゾーンについてテーブルを作成すると効率が良い。またRAM上に作成するテーブルは1ゾーンに制限されず、RAM容量が許さ

れるのであれば、複数ゾーン分作成してもかまわない。

【0112】

ホストからアクセス要求に対する変換テーブルの作成過程について以下に説明する。物理ブロックアドレス0～255を包括する領域を設定し、ゾーン1と呼ぶ。また物理ブロックアドレス256～511を包括する領域を設定し、ゾーン2と呼ぶことにする。ホストは現在RAM上に作成されている変換テーブルがどのゾーンのものであるかを覚えている。

【0113】

RAM上にゾーン1の変換テーブルが作成されており、論理ブロックアドレス128に対するアクセス要求が来た場合の手順を以下に説明する。

(1) 論理ブロックアドレス $128 - 250 \times (n-1) < 250$ となるような n を求めると $n=1$ である事から、ゾーン1の変換テーブルが必要であることが分かる。

(2) RAM上にある変換テーブルがゾーン1である為、テーブルの作成は行わない。

(3) 変換テーブル上のアドレスは0～249の値をとるため、ホストが要求した論理ブロックアドレス128から、変換テーブル上の参照すべきアドレスを求める。 $128 - 250 \times (1-1) = 128$ より、変換テーブル上のアドレス128に対応する物理ブロックアドレスに対しアクセスすればよい。

(4) 書き換えが発生した場合は変換テーブルを更新し、次のアクセスに備える。

【0114】

次に、RAM上にゾーン1の変換テーブルが作成されており、論理ブロックアドレス324に対するアクセス要求がきた場合の手順を以下に説明する。

(1) 論理ブロックアドレス $324 - 250 \times (n-1) < 250$ となるような n を求めると $n=2$ である事から、ゾーン2の変換テーブルが必要であることが分かる。

(2) RAM上にある変換テーブルがゾーン1である為、ゾーン2に包括された物理ブロックアドレス256～511の冗長部の論理ブロックアドレス情報をサ

ーチし、RAM上にゾーン2の変換テーブルの作成を行う。

(3) 変換テーブル上のアドレスは0～249の値をとるため、ホストが要求した論理ブロックアドレス324から、変換テーブル上の参照すべきアドレスを求める。 $324 - 250 \times (2-1) = 74$ より、変換テーブル上のアドレス74に対応する物理ブロックアドレスに対しアクセスすればよい。

(4) 書き換えが発生した場合は変換テーブルを更新し、次のアクセスに備える。

以上のように、アクセスする論理ブロックに応じて、変換テーブルを作成することで、従来より、RAM領域を容易に小さくすることが可能である。システムRAM上にゾーン2のテーブルが作成されている場合も、同様の過程で容易に目的のアドレスにアクセスすることができる。

【0115】

論理番地／物理番地変換テーブルを作成するときに、RAM上に論理ブロックと物理ブロックの変換テーブルを作成するが、このとき、作成するゾーンNO. が分かっているため、変換テーブル上の物理ブロックアドレスは0～255の値を取ればよい。実際に物理ブロックにアクセスするときは、OFFSETとして、変換テーブル上の物理アドレスに $256 \times \text{ゾーンNO.}$ を加える事で、容易にフラッシュメモリに入力すべき新の物理ブロックアドレスを得る事ができる。

【0116】

16MビットNAND型フラッシュメモリを用いた場合、従来制御方法では、物理ブロックアドレスを表すために9ビット必要であり、ソフトの利便性からは、2バイトを費やしていた。本実施例では、図35に示すように、物理ブロックアドレスを表すには8ビットあればよく、1バイトで済む。したがって、従来1kバイト必要であったRAM容量は半分に節約できる。論理ブロックアドレスが増加すると、当然、論理番地／物理番地変換テーブルの容量はそれにもなって増加するため、本実施例の効果はフラッシュメモリの容量が大きくなるほど、大きく貢献する。

【0117】

また、本実施例によれば、図11に示した物理ブロックの冗長部のBlock

Address Areaに格納可能なブロックアドレスで、大容量のフラッシュメモリシステムを制御可能とする。すなわち、1つのゾーンを上記冗長部のBlock Address Areaに格納可能なブロックアドレス値の物理ブロックで構成されるように分割すれば、大容量の論理アドレスに対応可能となる。

【0118】

図36は、16MビットNAND型フラッシュメモリを4つの均等な容量のゾーンに分割した場合の物理ブロックの構造を示した図である。

【0119】

ホストが該フラッシュメモリを制御する場合に、必要となる論理ブロック数は500個と定義され、論理ブロックアドレス0～499の値をとる。物理ブロックの冗長部には自分がどの論理ブロックに対応するデータを保持しているかを示す論理ブロックアドレス情報が格納されるが、本実施例では、論理ブロックアドレスは、ゾーン毎に0からシリーズに付与される。これらの論理ブロックアドレスに対応する物理ブロックアドレスは下記ようになる。ゾーン1には論理ブロックアドレス0～124が割り当てられ、それぞれの論理ブロックは物理ブロックアドレス0～127の128個の物理ブロックのいずれかに対応する。ゾーン2には論理ブロックアドレス125～249が割り当てられ、それぞれの論理ブロックは物理ブロックアドレス128～255の128個の物理ブロックのいずれかに対応する。ゾーン3には論理ブロックアドレス126～374が割り当てられ、それぞれの論理ブロックは物理ブロックアドレス256～383の128個の物理ブロックのいずれかに対応する。ゾーン4には論理ブロックアドレス384～499が割り当てられ、それぞれの論理ブロックは物理ブロックアドレス384～511の128個の物理ブロックのいずれかに対応する。

【0120】

データ更新時の追加書き込み方式は、書き込み要求の発生した論理ブロックが存在するゾーン内のみで行われ、ある論理ブロックアドレスに対応するデータが格納された物理ブロックアドレスは、固定ではなく、常にゾーン内で移動している。それぞれの物理ブロックの冗長部には自分がどの論理ブロックアドレスのデ

ータを保持しているかを示す論理ブロックアドレス情報を格納している。

【0121】

上述のように、各ゾーンには125個の論理ブロックが割り当てられ、また128個の物理ブロックが割り当てられる。以上のように物理ブロックを4つのゾーンに分割した場合も、2つのゾーンに分割した時同様の効果を有する。

【0122】

次に、本実施例のリダンダンシー動作について、上記物理ブロック4つのゾーンに分割した場合を例に説明する。

【0123】

本実施例の説明に先立って、ゾーン分割を行わない従来のフラッシュメモリのリダンダンシー動作について、16MビットNAND型フラッシュメモリを例に説明する。図2に示すように、16MビットのNAND型フラッシュメモリは512個の物理ブロックを有するが、図3に示すように、ホストから見た論理ブロックは500個に規定される。またPCカードATAインターフェースを容易に構築するための情報を格納するために1ブロック、追加書き込み方式を採用しているために1ブロック必要となる。従って本フラッシュメモリを制御する場合502個のブロックが必要となる。従って16MビットのNAND型フラッシュメモリでは10個までの不良ブロックの存在は許される。

【0124】

しかし、フラッシュメモリには書き換え回数の上限が存在し、使用中に不良ブロックが発生することがあり、十分な記憶領域を確保するために、不良ブロックが発生した場合の、置換先として、書き込み、消去可能な有効ブロックを持っておく必要がある。また、10個以上の不良ブロックを有するフラッシュメモリについては、出荷時の容量が十分に無いことから出荷できず、製品歩留まりを下げるという問題も発生する。

【0125】

このため、有効ブロック数を多く確保し、使用中の不良ブロックの発生に備えると共に、出荷時の製品歩留まりを向上させるために、16MビットNAND型フラッシュメモリには、512個の本体ブロックの他に複数の冗長ブロックが設

けられている。この冗長ブロックは出荷前に、冗長回路により、512個の本体ブロック中に発生している、不良ブロックと、ハード的に置き換えられる。冗長回路を使用すると、冗長ブロックは、不良ブロックのアドレスに割り付けられる。ハード的な置き換え後は、不良ブロックが存在したアドレスを選択すると、置き換えられた冗長ブロックが選択されるようになる。ここで、冗長ブロックは無限にあるわけではないため、出荷時に、置き換えきれなかった不良ブロックは、先天性不良ブロックとして扱われる。また、ユーザーの使用中に発生した不良ブロックは、後天性不良ブロックとして扱われる。この先天性不良ブロックと後天性不良ブロックは上記有効ブロックによって救済される。

【0126】

通常、本体ブロックと冗長ブロックとの置き換えを行う場合、冗長回路を使用し、ハード的に置き換えが行われるが、図37に示したように、置き換え作業の利便性から、ブロックアドレスの小さい不良ブロックから順に連続的に、あるいは、ブロックアドレスの大きい不良ブロックから順に連続的に置き換えが行われる。従って、ブロックの置き換え後は、ブロックアドレスの小さい方から、あるいはブロックアドレスの大きい方から連続的に書き込み、消去可能なブロックが存在している。

【0127】

このように、従来のフラッシュメモリの制御方法では、上述したように512個の物理ブロックに対し、論理ブロックが500個、PCカードATAインターフェースを容易に構築するための情報を格納するために1ブロック、追加書き込み方式を採用しているために空き領域として1ブロック必要であったため、都合、502個の書き換え可能なブロックがあれば制御可能であった。従って、従来の制御方法では、512個の物理ブロックに対し、10個の不良ブロックが許されていた。

【0128】

次に、本実施例のリダンダンシー動作について説明する。ゾーン1について考えると、PCカードATAインターフェースを容易に構築するための情報を格納するために1ブロック、追加書き込み方式を採用しているために空き領域として

1ブロック必要となる。従って、125個の論理ブロックに対し、127個の書き換え可能な物理ブロックが必要となり、ゾーン内で許される、書き換え不可能な不良ブロックは1個までとなる。ゾーン2～4については追加書き込み用の空きブロックとして、1ブロック余分に必要であることから、126個の書き換え可能な物理ブロックが必要となり、ゾーン内で許される、書き換え不可能な不良ブロックは2個までとなる。以下、説明を容易にするため、1～4ゾーンに対し、許される不良ブロック数はそれぞれ1個とする。従って、従来制御方法で許容された不良ブロック数のスペックに対し、非常に厳しくなる。

【0129】

図36に示す16MビットNAND型フラッシュメモリにおいて、例えば、ブロックアドレス2, 5, 129, 131, 132, 385, 389の7ブロックが不良ブロックであり、このフラッシュメモリは4個の冗長ブロックを有しているとする。図36に示すように、冗長回路を使用し、従来のようにブロックアドレスの小さい不良ブロックから順に、冗長ブロックと置き換えた場合、置き換え後の不良ブロックは物理ブロックアドレス132, 385, 389の3個のブロックとなり、これらは先天性ブロックとし、該ブロックの冗長部に先天性不良ブロックのマークがつけられる。ここで従来の制御方法では、先天性不良ブロックは10個まで許されたため、該フラッシュメモリを制御する上で特に問題はない。しかしゾーンに分割して制御することを前提とした場合、ゾーン4には物理ブロックアドレス385, 389の2個の不良ブロックが存在するため制御を行えず、この製品は出荷できなくなる。

【0130】

このように、本実施例のフラッシュメモリにおいて、不良ブロックがあった場合、本体の不良ブロックと冗長ブロックとの置き換えを、従来のフラッシュメモリ同様、単に置き換え作業の利便性から、ブロックアドレスの小さい不良ブロックから順に連続的に、あるいは、ブロックアドレスの大きい不良ブロックから順に連続的に置き換えを行うと、必要な良ブロックが確保できず、使用できないゾーンが発生する可能性が高くなる。

【0131】

このため、本実施例では、フラッシュメモリがゾーンに分割され制御される場合、全てのゾーンが有効ブロック数を満たすよう、各ゾーンに存在する不良ブロックと冗長ブロックとの置き換えを行う。

【0132】

図38は、ゾーン毎に存在する不良ブロックが、各ゾーンで許される不良ブロック数を越えないように、冗長ブロックとの置き換えを行った場合の物理ブロックの状態を示す図である。図38のフラッシュメモリも、図36に示したものの同様、ブロックアドレス2, 5, 129, 131, 132, 385, 389の7ブロックが不良ブロックと、4個の冗長ブロックを有している。従来の置き換え方法では出荷できなかったものが、良品として出荷できることが分かる。置き換え手順の一例を以下に説明する。

【0133】

(1) ゾーン1～ゾーン4までをサーチし、以下の変数の値を求める。

・不良物理ブロックアドレスの抽出を行う。

$Z(n) \text{ } BA(m)$ ($n=1\sim 4$ 、 $m=1\sim$ それぞれのゾーン内の不良ブロック数)

・ゾーン毎の不良ブロック数を求める。

$Z(n) \text{ } BN$ ($n=1\sim 4$)

(2) $Z(n) \text{ } BN$ の $n=1\sim 4$ のうち、1つでも冗長ブロックを全て使用しても、有効ブロック数を満たせないものがあれば、置換を行わず作業を終了する。

(3) $Z(n) \text{ } BA$ の $n=1\sim 4$ のうち、最も値の大きな n を抽出する。

(4) (3) で抽出された n について

$Z(n) \text{ } BA(m)$ ($m=1\sim$ ゾーン内の不良ブロック数) の内、ブロックアドレス情報が格納されている最も小さいか、大きい m について $Z(n) \text{ } BA(m)$ に格納された物理ブロックアドレスに該当するブロックを冗長ブロックと置き換える。

(5) $Z(n) \text{ } BN = Z(n) \text{ } BN - 1$

(6) (3) で選択された m について

Z(n) B A(m) のブロックアドレス情報を削除する。

(7) (3) ~ (6) を繰り返す。冗長ブロックを使い切れれば、(9) の処理へ移る。

(8) Z(n) B N (n=1 ~4) が等しければ最も小さいか、最も大きい n のゾーンについて (3) ~ (7) を繰り返す。

(9) Z(n) B N (n=1 ~4) をチェックし、スペックを越える n が存在すれば、不良品とする。

(10) 終了

【0134】

当然、本体ブロックと冗長ブロックとの置き換え手順は種々考えられる。本実施例では4つのゾーンに分割された場合を例に説明したが、ゾーンが、2つあるいは、奇数に分割されるケースも考えられる。また分割されたゾーンの容量（ブロック数）が異なる場合も考えられる。いずれの場合においても、本実施例は、本体ブロックと冗長ブロックを置き換える場合、置き換え作業後の各ゾーンに存在する不良ブロックの個数が、それぞれのゾーンで許される不良ブロック数を越えないよう、それぞれのゾーンに存在する不良ブロック数を監視しながら、冗長ブロックとの置き換えを行うところであり、本実施例の趣旨を逸脱しない範囲で、置き換えの手順は異なってもかまわない。

【0135】

以上に述べたように実施例によれば、論理ブロックとフラッシュメモリの物理ブロックが1対1で対応づけられているテーブルを使用する場合に比べ、アクセス対象となるゾーン単位でテーブルを作成するため、テーブルに必要とするRAM領域を低減することが可能で、これまで不可能であった汎用CPUの内蔵RAMのみでメモリを制御することが可能となり、外付けRAMを有していた従来のケースに対して大幅なコスト削減が可能となり、特に物理ブロック数の多い不揮発性半導体メモリの制御に効果が大きい。また、物理ブロックの冗長部のBlock Address Areaで決められたビット数で、大容量の論理アドレスに対応できる。

【0136】

また、本実施例によれば、複数の物理ブロックを論理的な複数のゾーンに割り振り、ゾーン毎に論理ブロックと物理ブロックの変換テーブルを作成し、メモリアクセスを行う制御方法を行うシステムに使用されるフラッシュメモリにおいて、本体の不良ブロックと冗長ブロックとを置き換える場合、従来アドレスの小さい（大きい）方から連続的に置き換えていたのに対し、本発明では、置き換え後の各ゾーンに存在する不良ブロックの個数が、それぞれのゾーンで許される不良ブロック数を越えないよう、それぞれのゾーンに存在する不良ブロック数を監視しながら、置き換えを行うため、使用不可能なゾーンを有するために出荷できなくなる製品が減り、製品歩留りが向上する。

【0137】

次に、本発明の第6の実施例について詳細に説明する。

本実施例は、2値／多値等のいずれのタイプにもサポート可能なフラッシュメモリカードについてである。

【0138】

図39は本実施例をPCカードアダプタに適応した図である。フラッシュメモリカード101Aは図17に示される2値対応のものであり、フラッシュメモリカード101Bは図18に示される4値対応のものである。PCカードアダプタ102は、フラッシュメモリカード101Aおよび101Bのデータをパソコンに転送するためのものである。PCカードアダプター102は、PCカードスロット用の68ピンのコネクタおよびフラッシュメモリカード用のコネクタを具備している。内部にはフラッシュメモリカードの制御およびPCカードスロットとの電気的なインタフェースを取るためのコントローラ103および、コントローラ内部のCPU用の発振子104、バッファ用のRAM105等がある。コントローラ103中には、本発明に直接的に関係する2種類のエラー訂正用の回路、ECC回路1およびECC回路2がある。

【0139】

図40に電源投入時の制御例を示す。

（ステップS1） 電源電圧検知

本メモリカードには、5V電源電圧と、3.3V電源電圧の製品があるが、3.3V電源電圧動作のメモリカードに5Vの電源が印加されると、電圧破壊等の問題が発生する可能性がある。これを回避するため、システムは電源電圧を検知している。

(ステップS2、3) IDチェック

フラッシュメモリカードには記憶容量またはインタフェース仕様の異なる複数種類の物がある。システムにメモリカードが挿入された時には、デバイスのメーカーコードおよびデバイスコード等を判別し、想定外のコードの場合は新たなアクセスはしないようにする。またメーカーコード、デバイスコード等の読み出しには正規の電源電圧を投入する。

(ステップS4～8) ECCのチェック

ここではECCの方式についてのチェックを行う。

【0140】

以下に、ECC方式を認識させる方法の一例を示す。第1の方法は、上記ステップS2、3と同様にデバイスコードにより判別する方法がある。例えば、同じ容量のフラッシュメモリであっても、メモリ構成によってデバイスコードを変えれば良い。

【0141】

図41は、本実施例の16Mビットフラッシュメモリのデータ部と冗長部の構成を示した図である。図4に示す従来のものと異なる点は、冗長部のUsre Data AreaがECC Flag AreaとECC Area-3とに割り当てられている点である。図41に示すようにECCの符号を格納する領域としてECC Area-1から3までの3個の領域を定義している。上記3個の領域の使用方法を図42を用いて説明する。ここでは、ECC Flag Areaバイト中にECCの方式に関する情報を記憶させておく。例えば、ECC方式1の場合Flagデータとしては“AAh”、ECC方式2の場合は“55h”と定義しておく。

【0142】

ECC方式1に基づく、フラッシュメモリーカードの場合、ECC Area

－1には、偶数ページデータ（256バイト）の3バイトECCコードが、ECC Area-2は、奇数ページデータ（256バイト）の3バイトECCコードが入り、ECC Area-3は空白である。ECC Area-1および2は各々256バイトのデータの内、2ビットのエラーを訂正できる。

【0143】

ECC方式2に基づく、フラッシュメモリーカードの場合、ECC Area-1、ECC Area-2、ECC Area-3に512バイトに対するECC符号が分散して記憶される。この場合512バイトのデータの内、2ビットのエラーを訂正できる。

【0144】

図39のPCカードアダプター102内のコントローラ103はフラッシュメモリーカードがアダプターに挿入されるとECC Flag Areaを読み出し、ECC方式を判別する。

【0145】

図39中では、ECC方式1およびECC方式2それぞれに対応した符号生成、およびエラー検出回路が選択される。コントローラ中のCPUはホストとバッファ用RAMとフラッシュメモリ間のデータの流れを制御する。また図39中のECC回路は、必ずしもハードウェアである必要性はない、ECC符号の生成等すべてをソフトウェアで実行することも勿論可能である。

【0146】

また本発明は上記実施例に限られない。発明の主旨を逸脱しない範囲で種々変更可能である。

【0147】

上記実施例では、2種類のECC方式の存在を仮定したが、方式は2種類には限られない。3種類以上のECC方式を設定可能である。

【0148】

また、ECC方式の選択には、ECCを使用しないという選択も含まれる。信頼性の非常に高いフラッシュメモリーカードや、音声分野等のデータで特に高い信頼性が必要の無い分野等ではECCは必須ではない。その場合、上記ECC F

lag Areaの内容が“FFh”であればECCを使用していないと定義しても良い。

【0149】

また上記実施例では、フラッシュメモリ毎にECCの方式を定義したが、本発明はそれにかぎられない。例えば、セクター単位、もしくはブロック単位等、任意の単位でECCの方式を切り替えてもよい。この場合、単純に電源投入時にECCの方式を判別するのではなく前述の所定単位にアクセスする都度ECCの方式を切り替えてもよい。また例えば、ECCの方式1で読み出されたデータを方式2に変換して再格納するような動作も想定可能である。

【0150】

また上記実施例では、PCカードアダプターを例に説明したが、本実施例はPCカードアダプターに限られない。例えば、デジタルスチルカメラやPDA、ワードプロセッサ、音声録音機等各種機器に対しても同様に議論可能である。このように本実施例を使用すれば、非常に広い使用範囲のフラッシュメモリカードを1つのシステムで使用可能となる。

【0151】

以上に述べたように、本実施例によれば、エラー訂正アルゴリズムの異なる複数のフラッシュメモリカードを一つのシステムが取り扱い可能となるので、飛躍的に汎用性が向上する。

【0152】

【発明の効果】

本発明によれば、電子機器のコネクタに対して記憶媒体を挿入／抜去する際に、両者のいわゆる種々のピンの当接、離反に一定の順序をもたせるようにしたので、動作の安定性の向上とデータの保護が確実に行える。

【0153】

さらに、本発明によれば、論理アドレスと物理アドレスとを変換テーブルで変換するに際し、1個の論理ブロックに対して複数個の物理ブロックを割り振るようにしたので、テーブルが必要とするRAM領域を低減することが可能となる。

【0154】

さらに本発明によれば、クラスタの区切れが消去単位となるブロックをまたがないようにしたので、データ書換え時に発生する消去回数及び書き込み回数を減らすことができる。

【0155】

さらに、本発明によれば、消去コマンド実行時に物理ブロックの管理領域を開放すると同時に、そのデータ領域の消去も行うことにしたので、その後の書込みコマンド実行時の処理速度の向上することが可能となる。

【0156】

さらに本発明によれば、論理ブロックと物理ブロックの対応関係のうちの必要な領域の対応関係のみを逐次作成するようにしたので、最低限必要なRAM領域を低減することができる。

【0157】

さらに、本発明によれば、不良ブロックを冗長ブロックに置き換えるに際し、論理的なゾーンのそれぞれにおいて、置き換え後には、各ゾーンに存在する不良ブロック数が所定値を越えないようにしたので、製品の歩留りを向上させることができる。

【0158】

さらに本発明によれば、使用する記憶媒体の種別に応じてエラー訂正アルゴリズムを選択するようにしたので、各種の記憶媒体を使用でき、汎用性が向上する。

【図面の簡単な説明】

【図1】

メモ리카ード外觀図である。

【図2】

フラッシュメモリの物理ブロック構成を示す図である。

【図3】

パソコンにおけるデータを示す図である。

【図 4】

フラッシュメモリのデータ格納方法を示す図である。

【図 5】

フラッシュメモリの物理ブロック構成を示す図である。

【図 6】

フラッシュメモリの物理ブロック構成を示す図である。

【図 7】

フラッシュメモリのデータ格納方法を示す図である。

【図 8】

従来例の論理・物理ブロック変換テーブルを示す図である。

【図 9】

従来のメモリシステムにおける論理アドレスと物理アドレスの変換テーブル構成を示す図である。

【図 10】

64MビットNAND型非フラッシュメモリシステムにおける論理アドレスと物理アドレスの変換テーブル構成を示す図である。

【図 11】

フラッシュメモリのブロックアドレスのデータ配置を示す図である。

【図 12】

従来のDOSフォーマットのパラメータを示す図である。

【図 13】

従来の書き替えシーケンスを示す図である。

【図 14】

従来の書き替えシーケンスを示す図である。

【図 15】

従来の書き替えシーケンスを示す図である。

【図 16】

従来の消去コマンド実行時の管理領域とデータ領域の関係を示した図である。

【図 17】

フラッシュメモリの格納データ例を示す図である。

【図 18】

フラッシュメモリの格納データ例を示す図である。

【図 19】

従来のシステムを示す図である。

【図 20】

本発明のメモリシステムにおける ECC データ構成を示す図である。

【図 21】

本発明のメモリシステムにおける ECC コードを算出するための条件を示した図である。

【図 22】

本発明のメモリシステムにおける電源投入時の制御フローチャートである。

【図 23】

本発明のメモリシステムにおける論理アドレス／物理アドレス変換テーブル作成のフローチャートである。

【図 24】

本発明のメモリシステムにおける論理アドレスと物理アドレスの変換テーブル構成を示す図である。

【図 25】

本発明のメモリシステムにおける読み出し時のフローチャートである。

【図 26】

本発明のメモリシステムにおける書き込み時のフローチャートである。

【図 27】

本発明の DOS フォーマットのパラメータを示す図である。

【図 28】

本発明の書き換えシーケンスを示す図である。

【図 29】

本発明の書き換えシーケンスを示す図である。

【図 30】

本発明の書き換えシーケンスを示す図である。

【図 31】

本発明の消去コマンド実行時の管理領域とデータ領域の関係を示した図である。

【図 32】

本実施例による論理ブロック-物理ブロック変換テーブル例を示す図である。

【図 33】

本実施例による論理ブロック-物理ブロック変換テーブル作成例を示す図である。

【図 34】

実施例の論理・物理ブロック変換テーブルを示す図である。

【図 35】

実施例での物理ブロックアドレスの表現方法を示す図である。

【図 36】

従来の不良ブロックと冗長ブロックの置換えを行った場合の実施例を説明するための図である。

【図 37】

不良ブロックと冗長ブロックの置き換えの従来例を説明するための図である。

【図 38】

ゾーン分割制御を前提に不良ブロックと冗長ブロックの置き換えを行った場合の実施例を説明するための図である。

【図 39】

本発明の PC カードアダプタの構成を示す図である。

【図 40】

フラッシュメモリカードを PC カードアダプタに挿入したときの、フローチャートを示す図である。

【図 4 1】

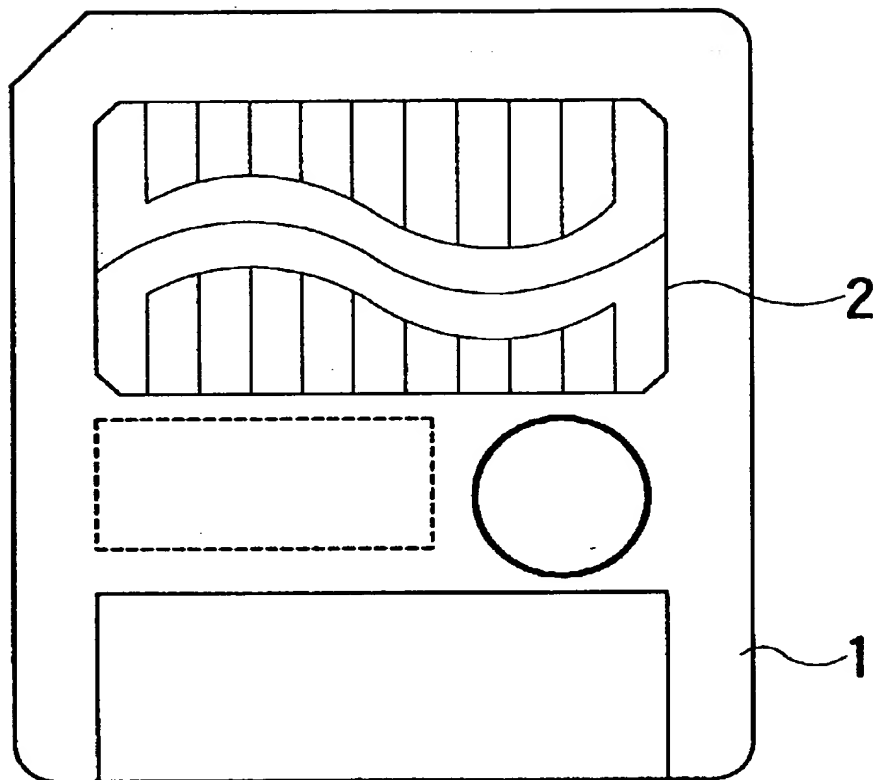
本発明の第 7 の実施例に係るフラッシュメモリの DOS フォーマットのパラメータを示す図である。

【図 4 2】

ECC 符号領域の使用方法を示す図である。

【書類名】 図面

【図 1】



【図 2】

		0	255	256	263
物理 ブロック 0	Page 0	データ領域(256バイト)		冗長部(8バイト)	
	Page 1				
	⋮				
	Page 15				
物理 ブロック 1	Page 0				
	Page 1				
	⋮				
	Page 15				
⋮	⋮	⋮		⋮	
物理 ブロック 511	Page 0				
	Page 1				
	⋮				
	Page 15				

【図 3】

物理 ブロック 0	セクタ 0	512バイト
	セクタ 1	
	⋮	
	セクタ 7	
物理 ブロック 1	セクタ 8	
	セクタ 9	
	⋮	
	セクタ 15	
⋮	⋮	⋮
物理 ブロック 499	セクタ 3992	
	セクタ 3993	
	⋮	
	セクタ 3999	

【図 4】

データ部

バイト	0ページ (偶数ページ)	1ページ (奇数ページ)
0~255	DATA Area-1	DATA Area-2

冗長部

バイト	偶数ページ	奇数ページ
256	User Data Area	ECC Area-2
257		
258		
259		
260	Data Status Area	Block Address Area-2
261	Block Status Area	
262	Block Address Area-1	ECC Area-1
263		

【図5】

		0	511	512	527
物理 ブロック 0	Page 0	データ領域(512バイト)		冗長部(16バイト)	
	Page 1				
	⋮				
	Page 15				
物理 ブロック 1	Page 0				
	Page 1				
	⋮				
	Page 15				
⋮	⋮	⋮	⋮	⋮	⋮
物理 ブロック 1023	Page 0				
	Page 1				
	⋮				
	Page 15				

【図 6】

論理 ブロック 0	セクタ 0	512バイト
	セクタ 1	
	⋮	
	セクタ 15	
論理 ブロック 1	セクタ 16	
	セクタ 17	
	⋮	
	セクタ 31	
⋮	⋮	⋮
論理 ブロック 999	セクタ 15984	
	セクタ 15985	
	⋮	
	セクタ 15999	

【図 7】

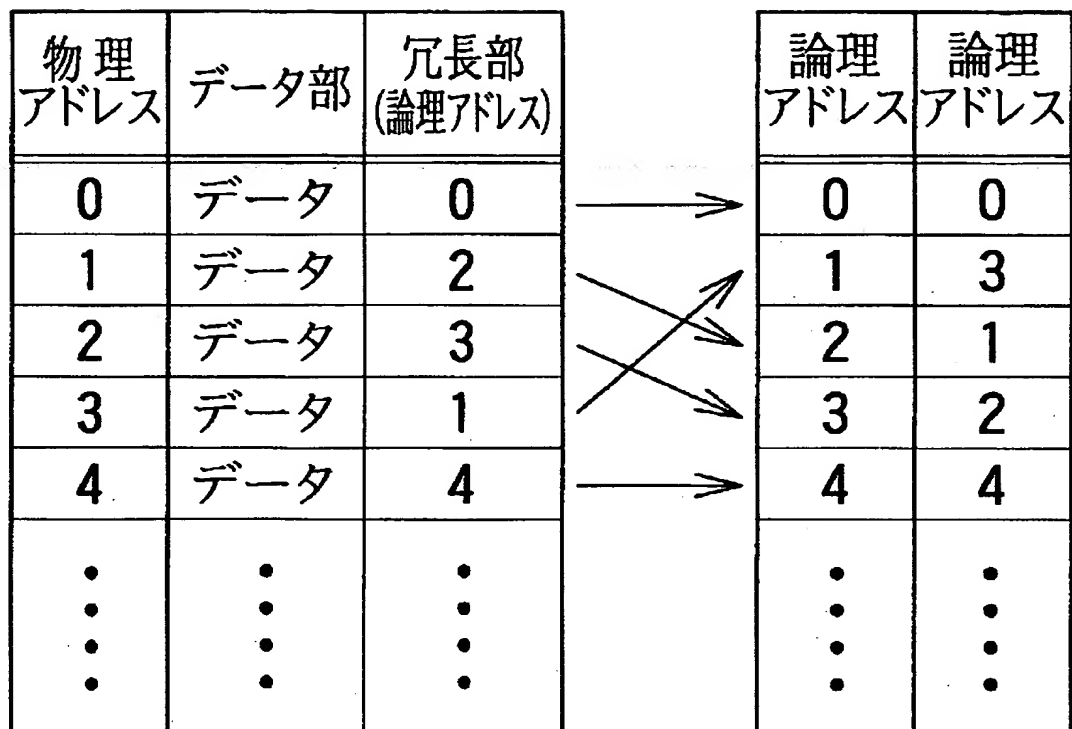
データ部

バイト	
0~511	DATA Area

冗長部

バイト	
512	User Data Area
513	
514	
515	
516	Data Status Area
517	Block Status Area
518	Block Address Area-1
519	
520	ECC Area-2
521	
522	
523	Block Address Area-2
524	
525	ECC Area-1
526	
527	

【図 8】



【図 9】

OFFSET (論理 ブロック アドレス)	物理 ブロック アドレス	物理ブロックアドレス (2進データ)			
		上位バイト		下位バイト	
word0 (LBA=0)	0	0000	0000	0000	0000
word1 (LBA=1)	500	0000	0001	1111	0100
word2 (LBA=2)	327	0000	0001	0100	0111
⋮	⋮	⋮	⋮	⋮	⋮
word497 (LBA=497)	244	0000	0000	1111	0100
word498 (LBA=498)	249	0000	0001	1110	1111
word499 (LBA=499)	128	0000	0001	1000	0000

【図 10】

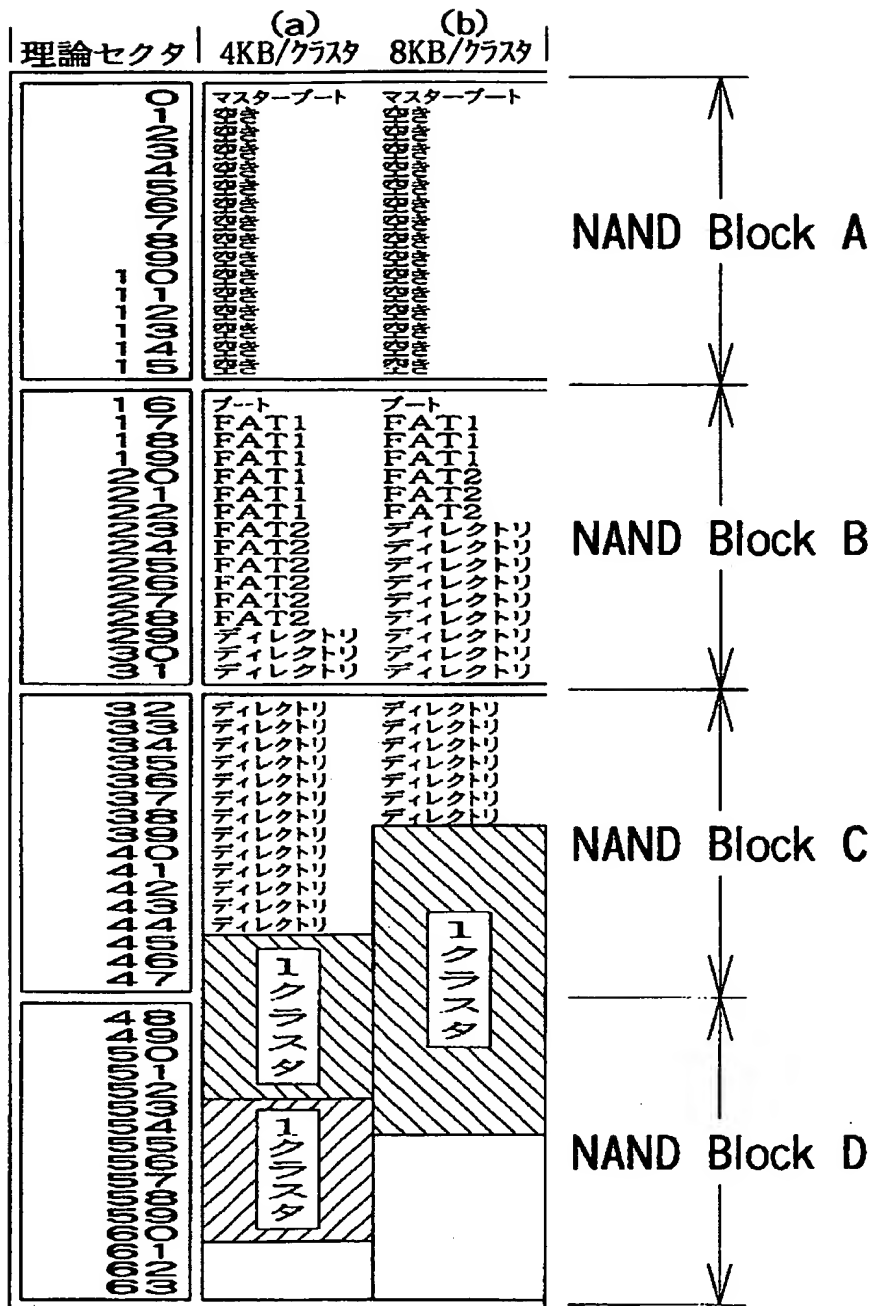
OFFSET (論理 ブロック アドレス)	物理 ブロック アドレス	物理ブロックアドレス (2進データ)			
		上位バイト		下位バイト	
word0 (LBA=0)	0	0000	0000	0000	0000
word1 (LBA=1)	1000	0000	0011	1110	1000
word2 (LBA=2)	654	0000	0010	1000	1110
⋮	⋮	⋮	⋮	⋮	⋮
word997 (LBA=997)	488	0000	0001	1110	1000
word998 (LBA=998)	498	0000	0001	1111	0010
word999 (LBA=999)	256	0000	0001	0000	0000

【図 1 1】

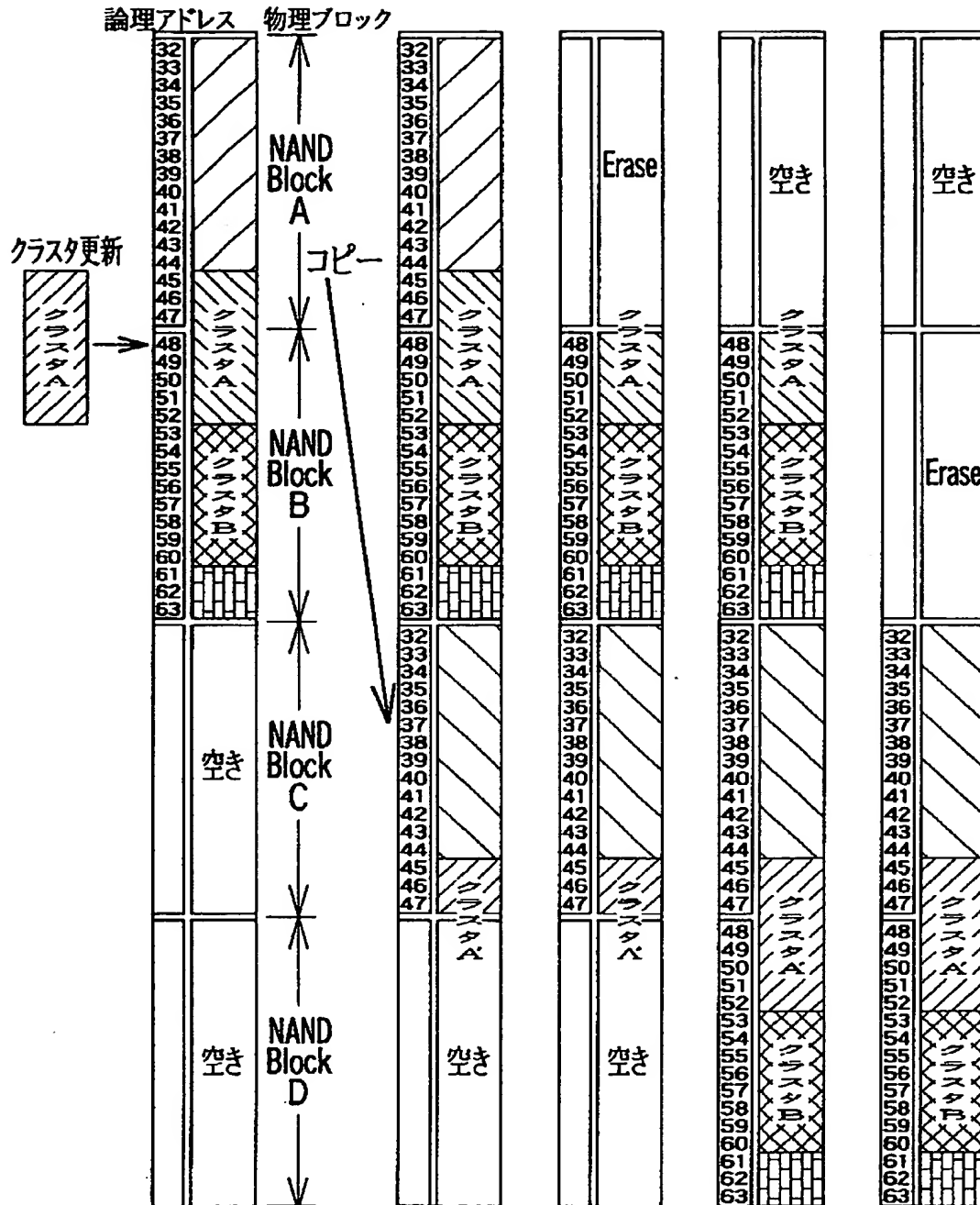
D7	D6	D5	D4	D3	D2	D1	D0	256+8byte/page
0	0	0	1	BA10	BA9	BA8	BA7	262 Byte (偶数ページ) 259 Byte (奇数ページ)
BA6	BA5	BA4	BA3	BA2	BA1	BA0	P	263 Byte (偶数ページ) 260 Byte (奇数ページ)

BA10~BA0: Logical Block Address P: 偶数パリティビット"1"固定値

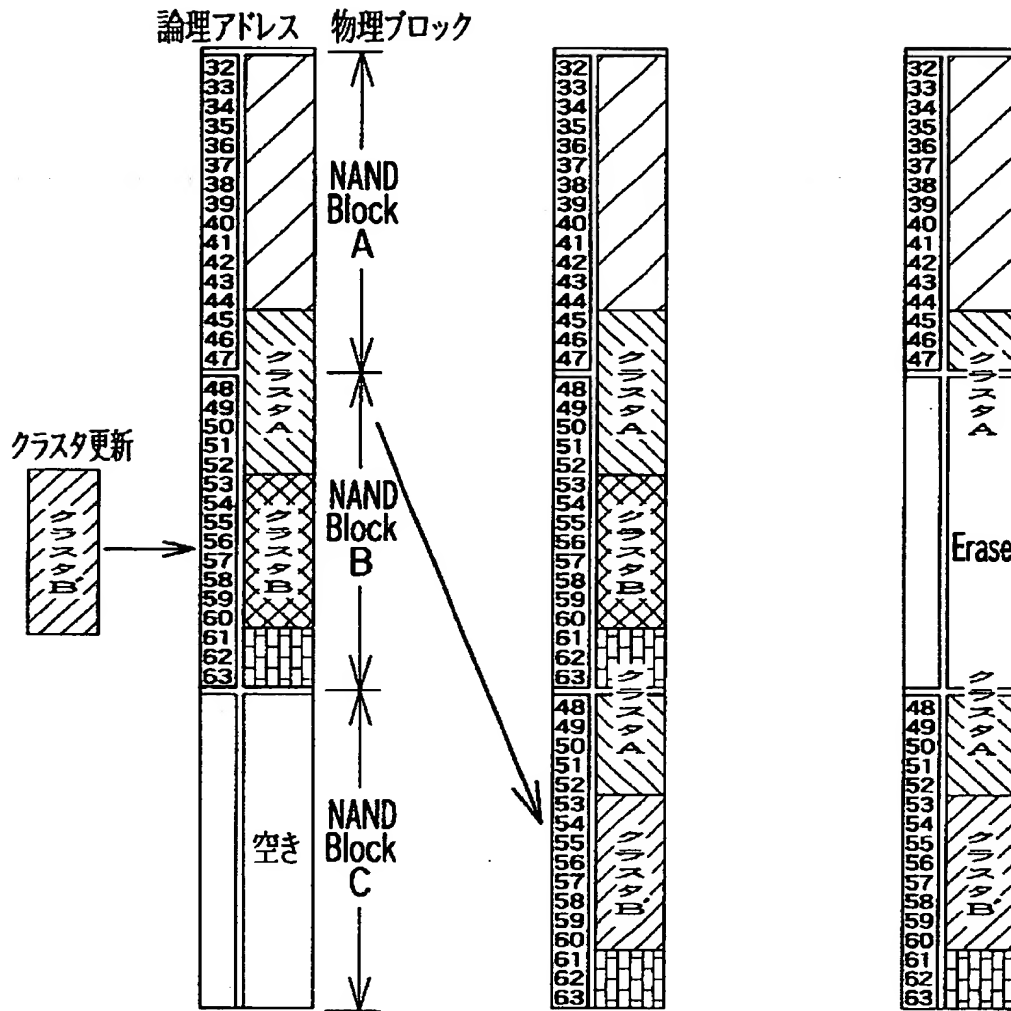
【図 1 2】



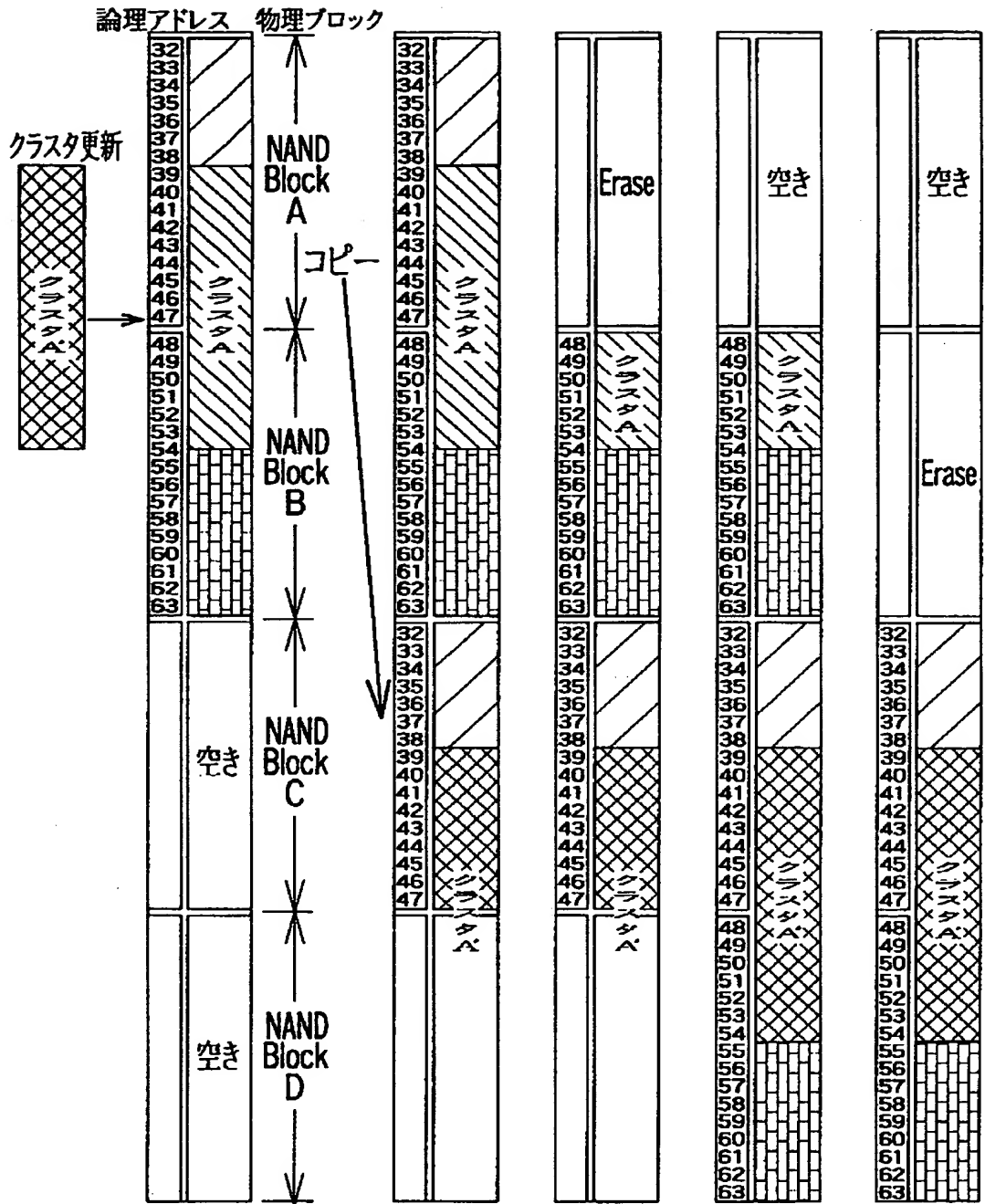
【図 13】



【図 14】



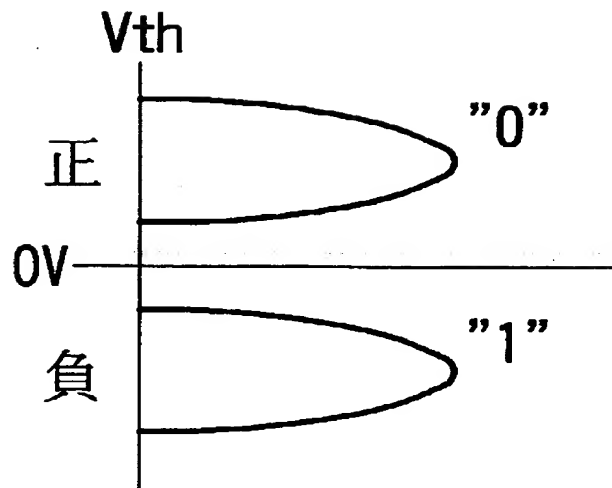
【図 15】



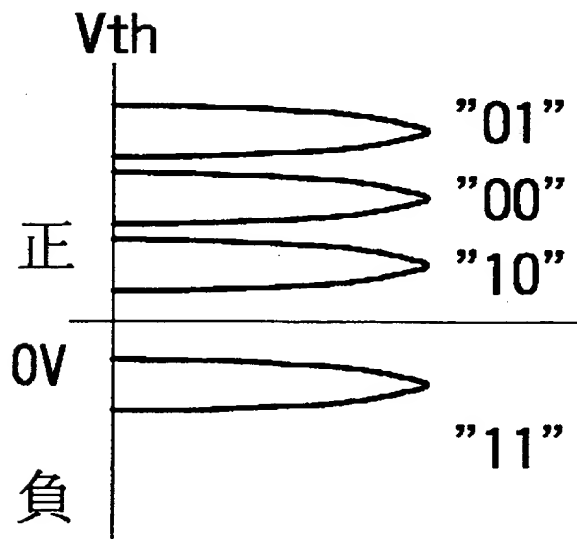
【図 16】



【図 1 7】



【図 1 8】



【図 19】

図2(a)のカード	図2(b)のカード
<div data-bbox="337 1129 669 1543"> <div data-bbox="375 1354 594 1514">CPU</div> <div data-bbox="521 1157 634 1333">ECC 回路1</div> </div> <div data-bbox="673 1276 734 1535">システムA</div>	使用不可
<div data-bbox="797 1129 1128 1543"> <div data-bbox="834 1354 1053 1514">CPU</div> <div data-bbox="980 1157 1094 1333">ECC 回路2</div> </div> <div data-bbox="1133 1276 1193 1535">システムB</div>	使用可能

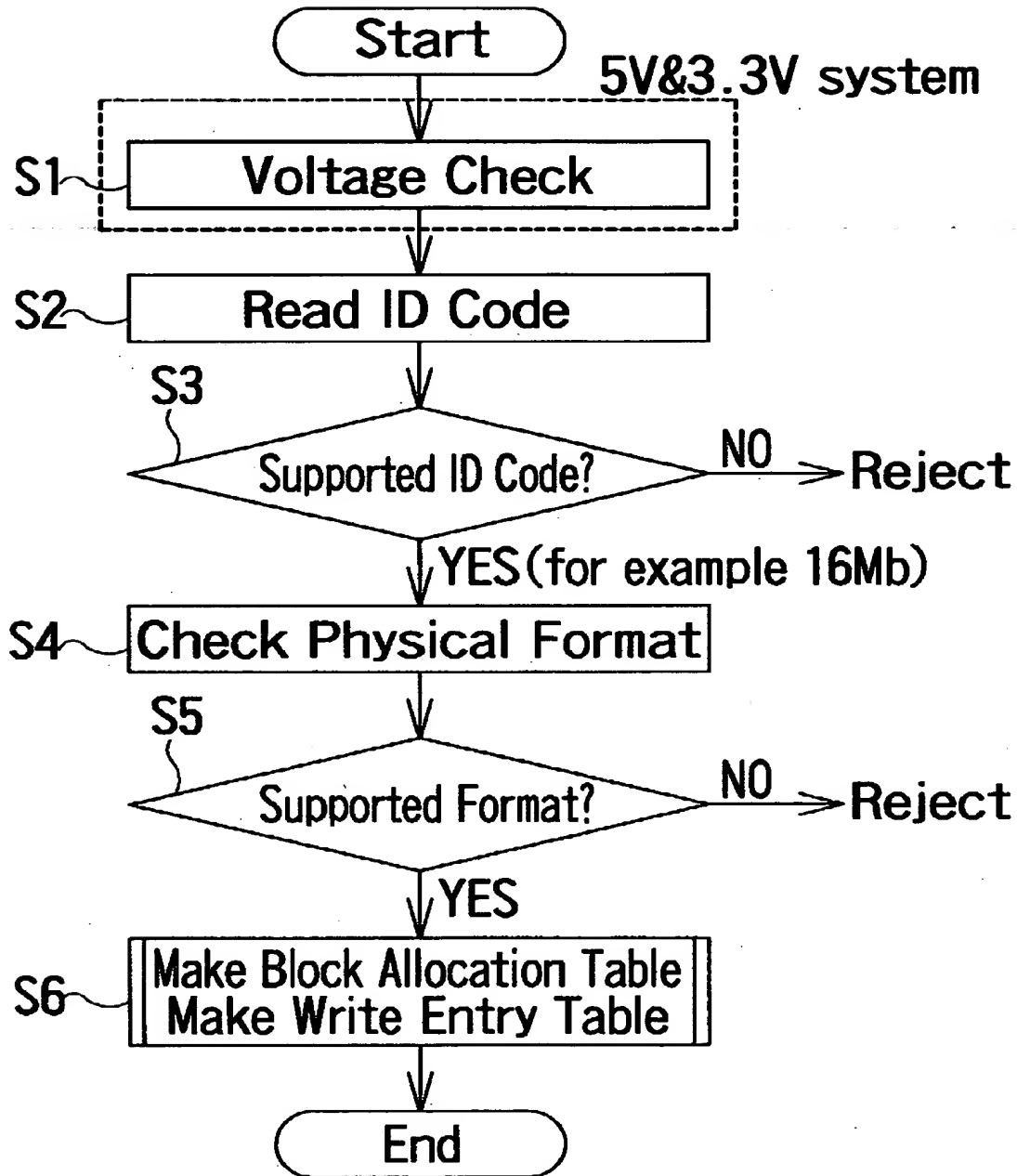
【図 20】

bit7	bit6		bit1		bit0	
1stByte	00000000	111	00000000	110	00000000	001
2ndByte	00000000	111	00000001	110	00000001	001
⋮						
255thByte	11111110	111	11111110	110	11111110	001
266thByte	11111111	111	11111111	110	11111111	001

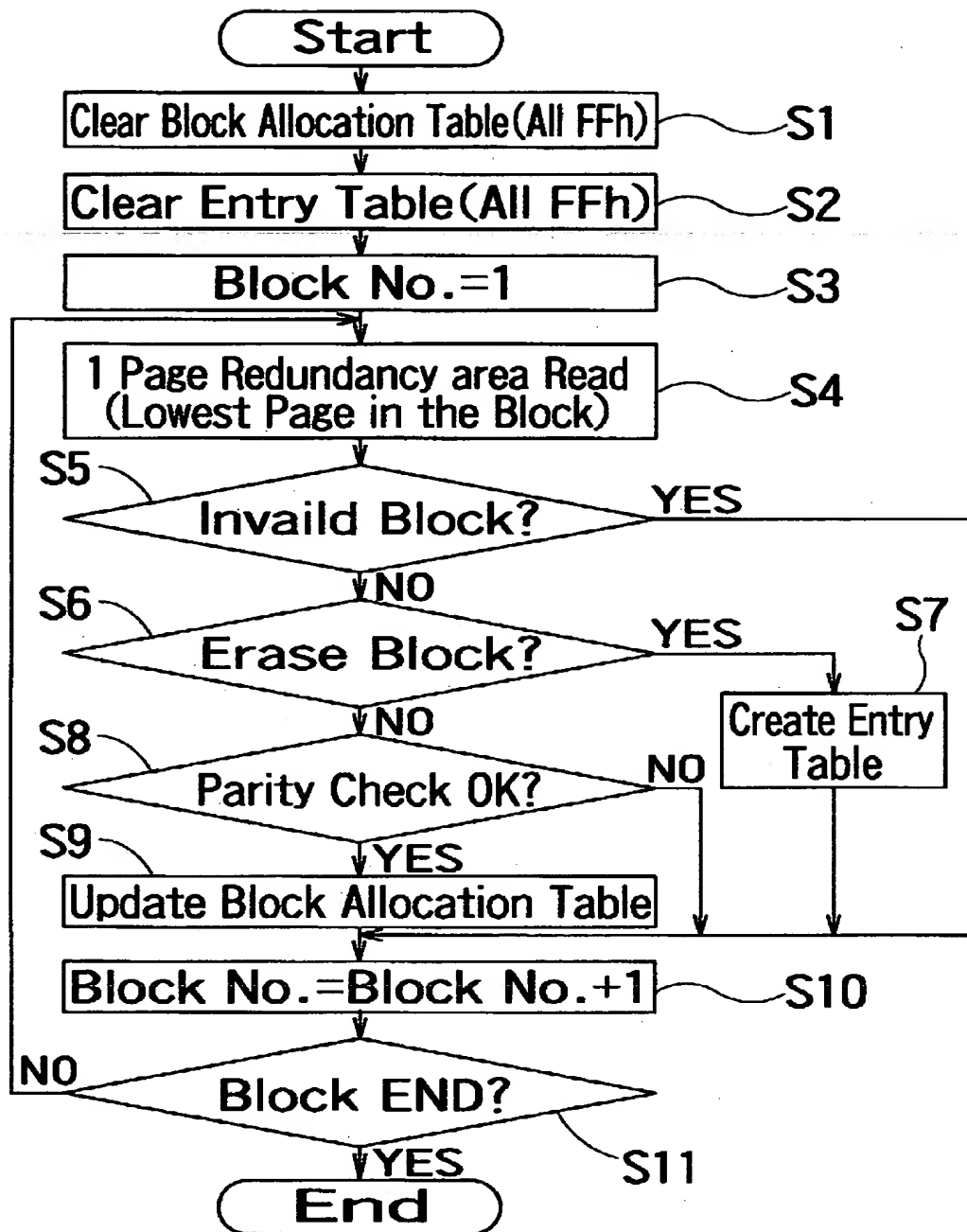
【図 21】

LP00=D(*****0、***)、LP01=D(*****1、***)
 LP02=D(*****0*、***)、LP03=D(*****1*、***)
 LP04=D(*****0**、***)、LP05=D(*****1**、***)
 LP06=D(****0***、***)、LP07=D(****1***、***)
 LP08=D(**0****、***)、LP09=D(**1****、***)
 LP010=D(**0*****、***)、LP011=D(**1*****、***)
 LP012=D(*0*****、***)、LP013=D(*1*****、***)
 LP014=D(0*****、***)、LP015=D(1*****、***)
 LP00=D(*****、**0)、LP01=D(*****、**1)
 LP02=D(*****、*0*)、LP03=D(*****、*1*)
 LP04=D(*****、0**)、LP05=D(*****、1**)

【図 22】



【図 23】

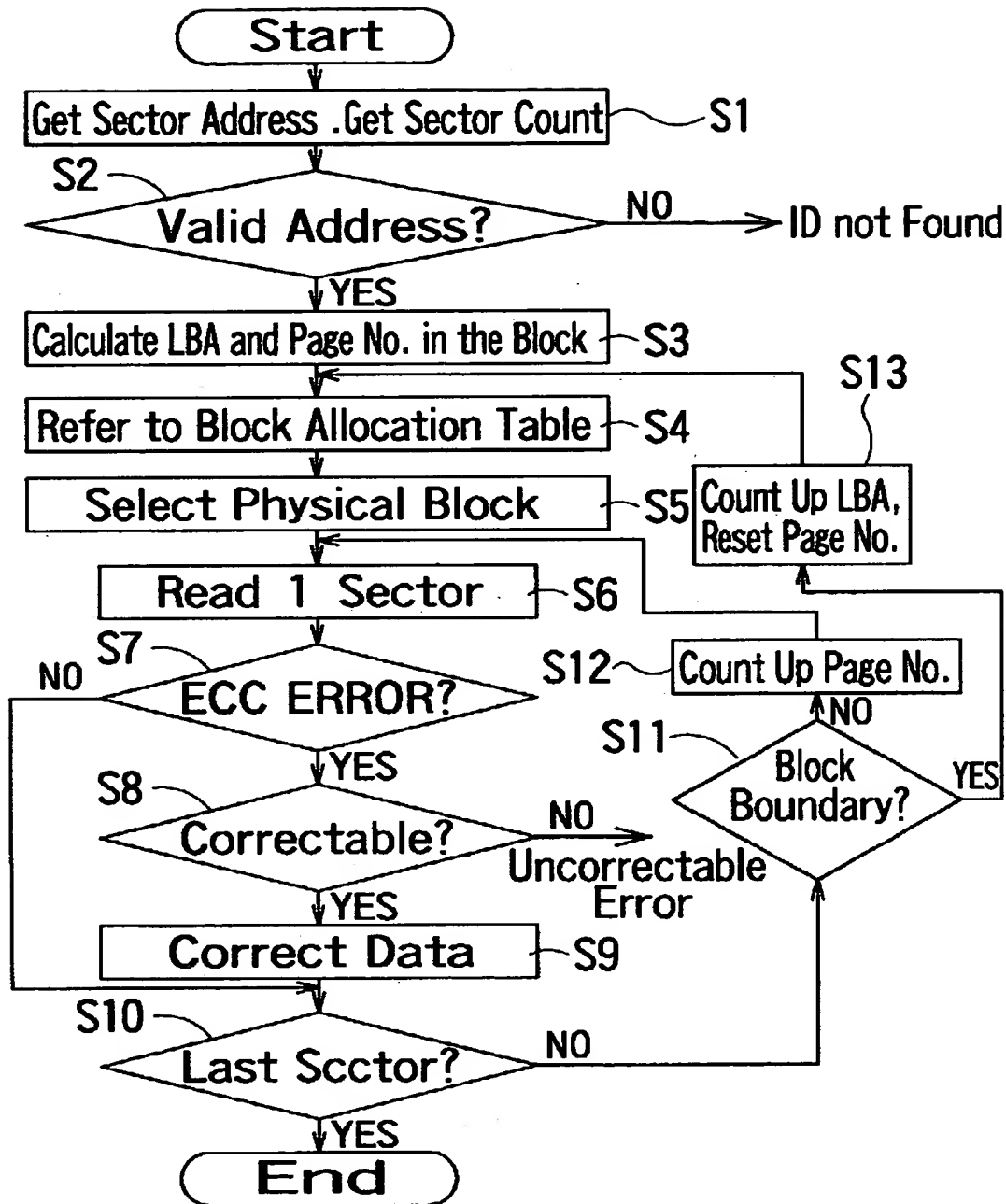


【図 24】

OFFSET (論理 ブロック アドレス)	物理 ブロックエリア アドレス	物理ブロック エリアアドレス (2進データ)	
Word0(LBA=0)	0	0000	0000
Word1(LBA=1)	250	1111	1010
Word2(LBA=2)	163	1010	0011
⋮	⋮	⋮	⋮
Word497(LBA=497)	122	0111	1010
Word498(LBA=498)	248	1010	1000
Word499(LBA=499)	64	0100	0000

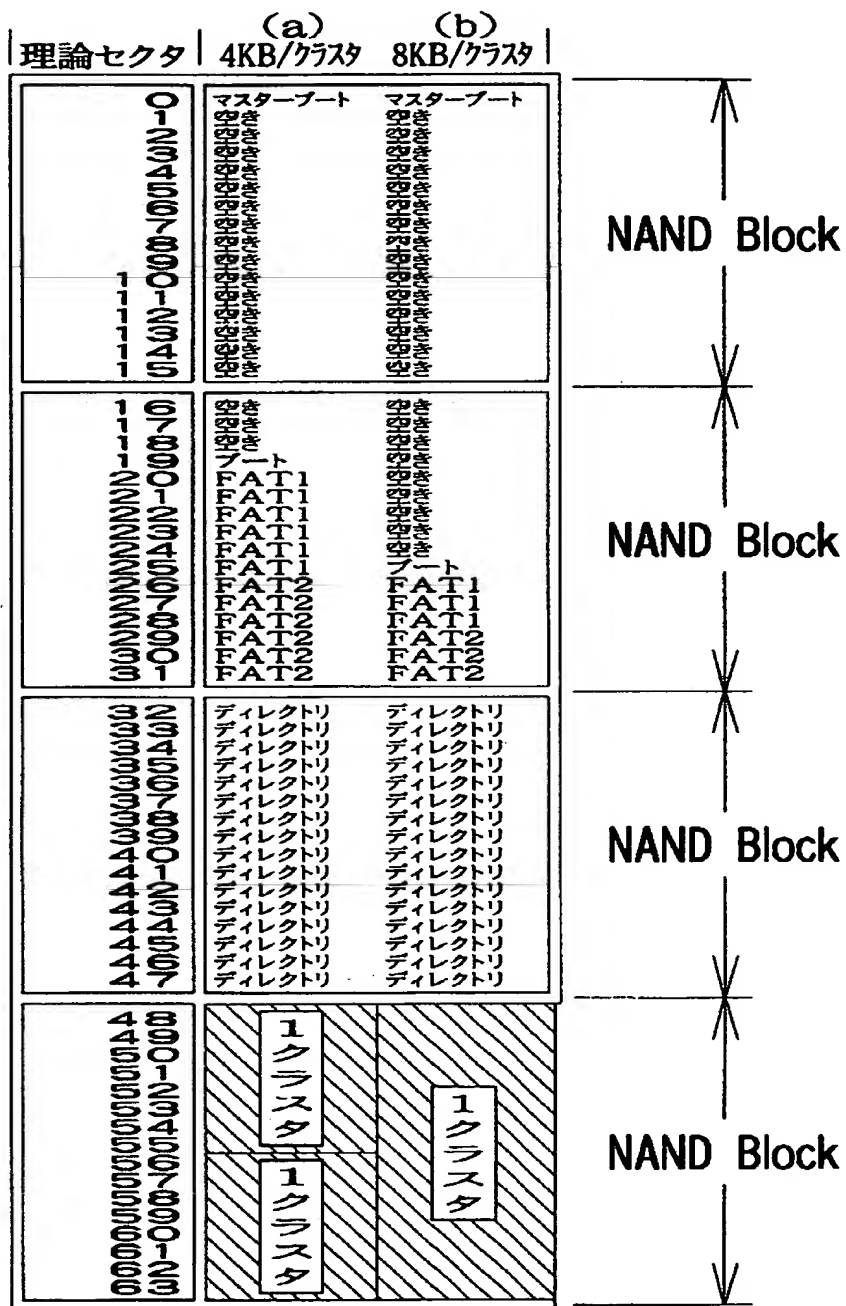
1物理ブロックエリア=2物理ブロック

【図 25】

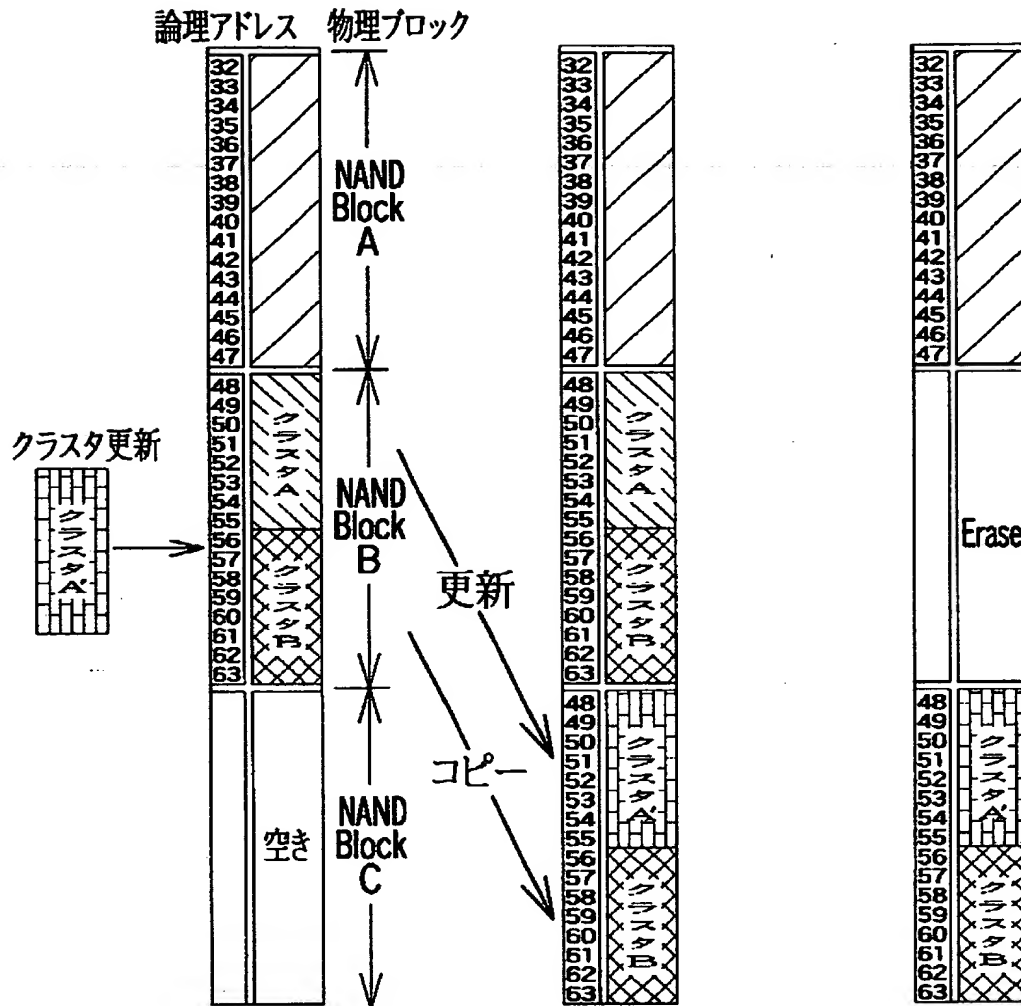




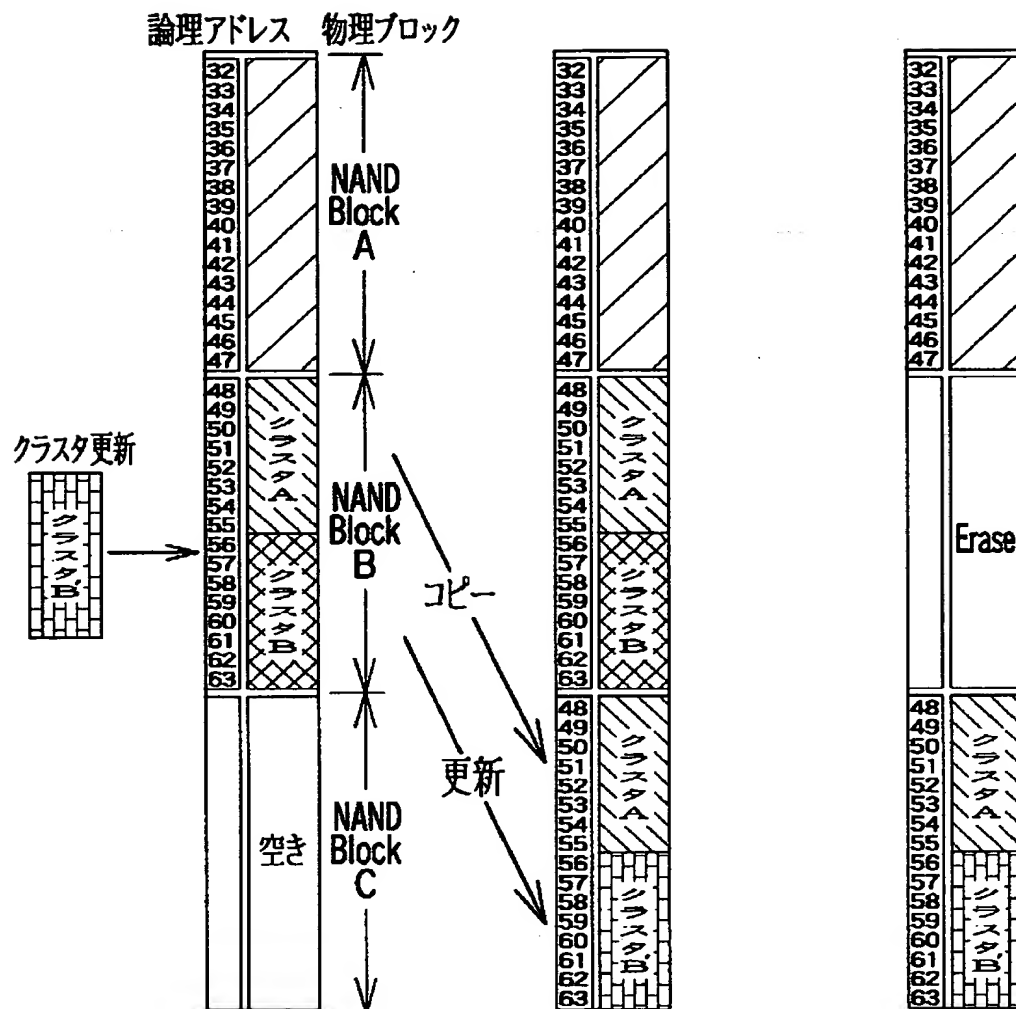
【图 2 7】



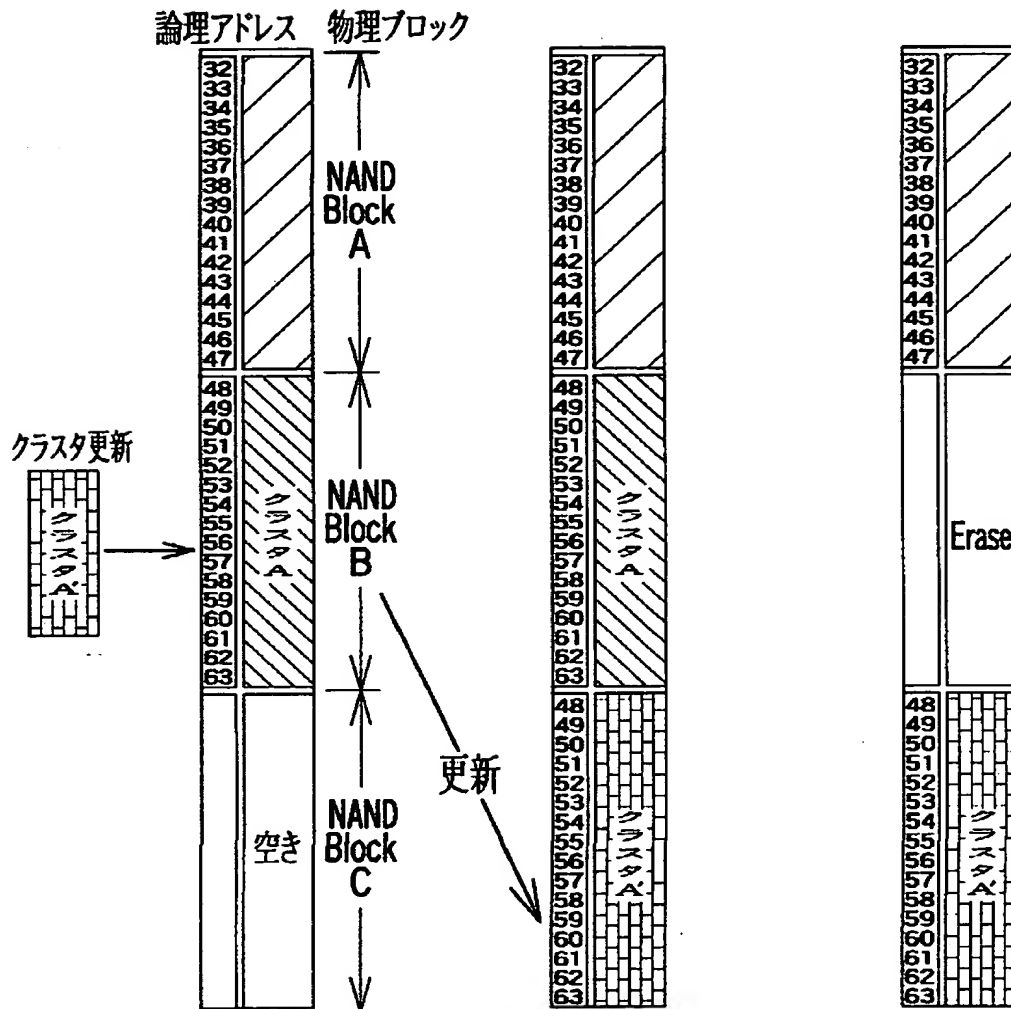
【図 28】



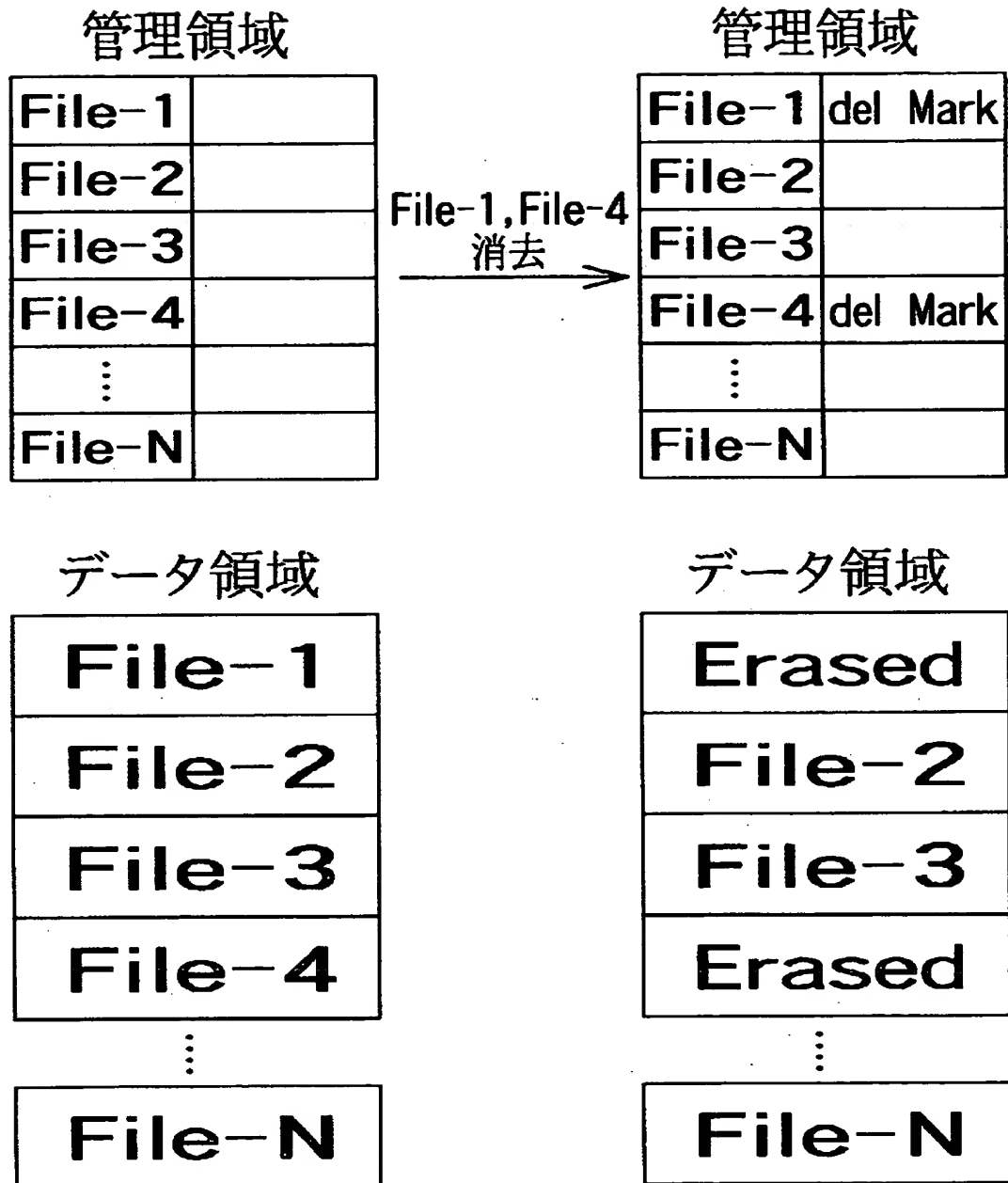
【図 29】



【図 30】



【図 31】



【図 3 2】

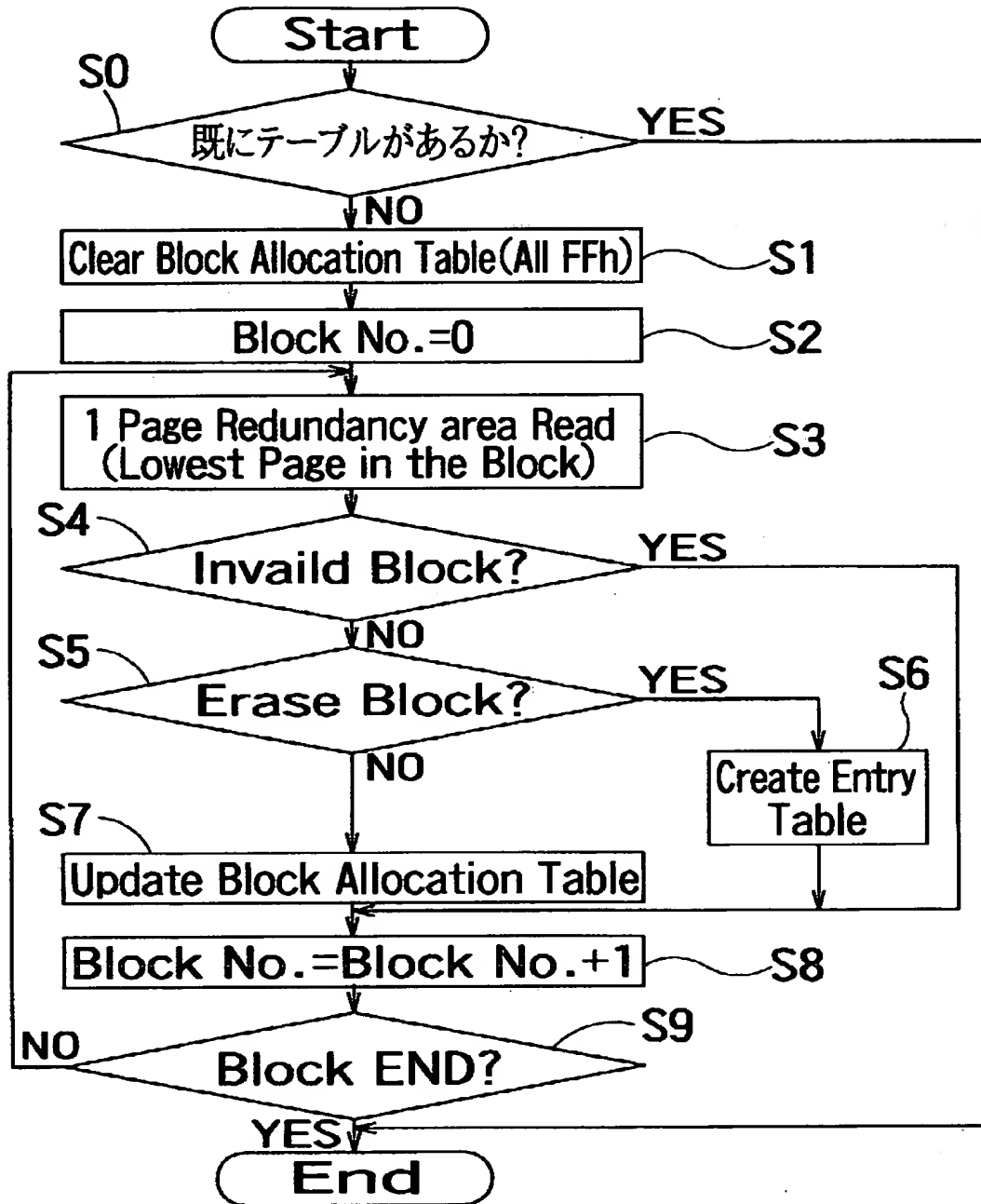
OFFSET (論理ブロックアドレス)	(物理ブロックアドレス)	
	Upper Byte	Lower Byte
Word0 (LBA=0)	Physical Block Address 上位	Physical Block Address 下位
Word1 (LBA=1)	Physical Block Address 上位	Physical Block Address 下位
Word2 (LBA=2)	Physical Block Address 上位	Physical Block Address 下位
⋮		
Word247 (LBA=247)	Physical Block Address 上位	Physical Block Address 下位
Word248 (LBA=248)	Physical Block Address 上位	Physical Block Address 下位
Word249 (LBA=249)	Physical Block Address 上位	Physical Block Address 下位

(a)

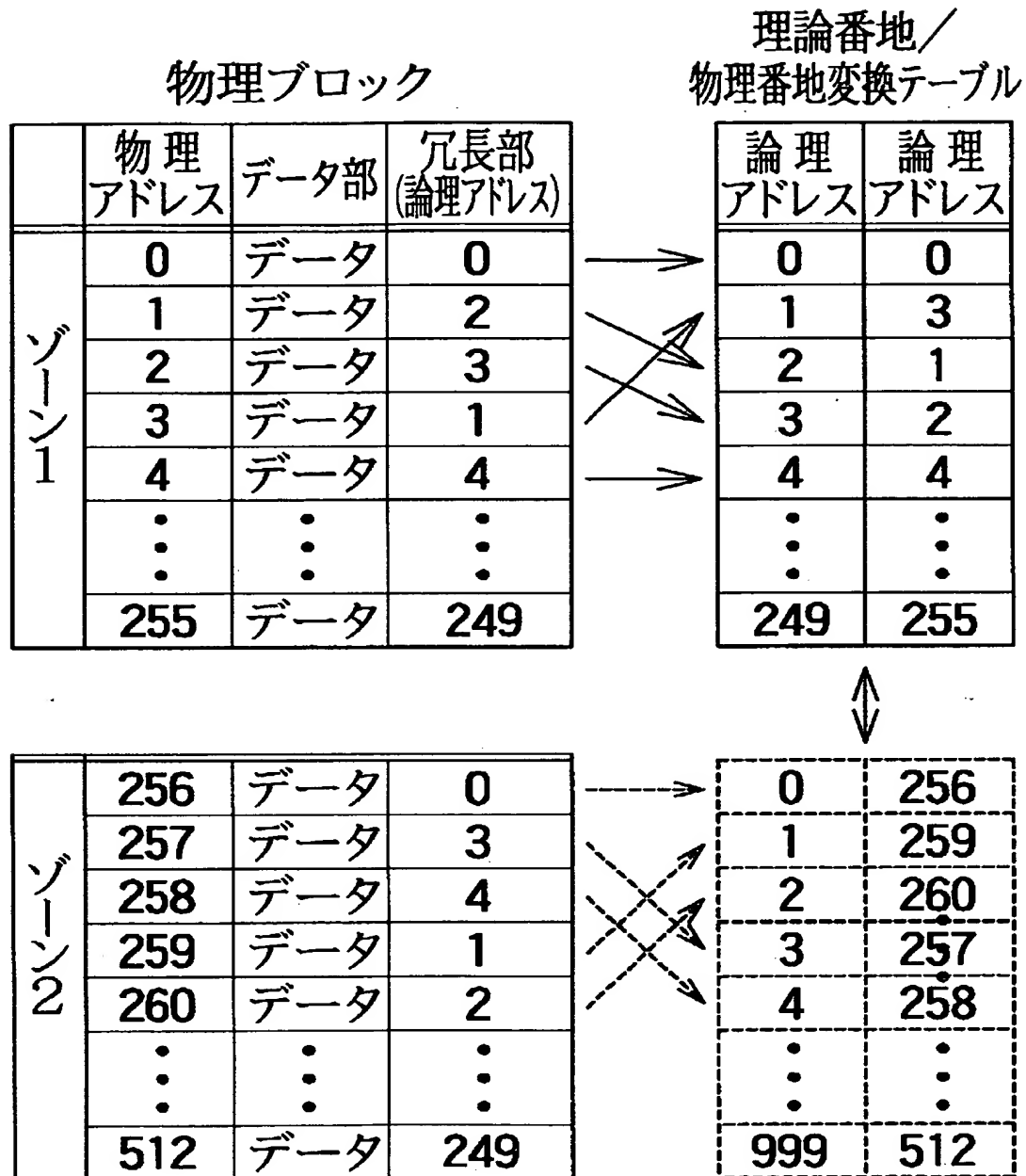
OFFSET (論理ブロックアドレス)	(物理ブロックアドレス)	
	Upper Byte	Lower Byte
Word0 (LBA=250)	Physical Block Address 上位	Physical Block Address 下位
Word1 (LBA=251)	Physical Block Address 上位	Physical Block Address 下位
Word2 (LBA=252)	Physical Block Address 上位	Physical Block Address 下位
⋮		
Word247 (LBA=497)	Physical Block Address 上位	Physical Block Address 下位
Word248 (LBA=498)	Physical Block Address 上位	Physical Block Address 下位
Word249 (LBA=499)	Physical Block Address 上位	Physical Block Address 下位

(b)

【図 33】



【図 34】



【図 3 5】

offset (論理ブロック アドレス)		物理ブロック アドレス	物理ブロックアドレス (2進データ)		
ゾーン1	Word0 (LBA=0)	0	0000	0000	0000
	Word2 (LBA=2)	227	0000	1110	0011
	⋮	⋮	⋮	⋮	⋮
	Word254 (LBA=254)	244	0000	1111	0100
	Word255 (LBA=255)	128	0000	1000	0111
ゾーン2	Word256 (LBA=256)	256(256-256=0)	0000	0000	0000
	Word257 (LBA=257)	327(327-256=71)	0000	0100	0111
	⋮	⋮	⋮	⋮	⋮
	Word499 (LBA=499)	500(500-256=244)	0000	1110	0000
	Word500 (LBA=500)	428(428-256=172)	0000	1010	1100

【図 3 6】

ブロック置き換え前			ブロック置き換え後		
物理ブロック アドレス			物理ブロック アドレス		
NG ゾ ー ン 1	0	Block0 data area	NG ゾ ー ン 1	0	Block0 data area
	1	Block1 data area		1	Block1 data area
	2	Block2 data area		2	Block2 data area
	3	Block3 data area		3	Block3 data area
	4	Block4 data area		4	Block4 data area
	5	Block5 data area		5	Block5 data area
	6	Block6 data area		6	Block6 data area
NG ゾ ー ン 2	128	Block128 data area	NG ゾ ー ン 2	128	Block128 data area
	129	Block129 data area		129	Block129 data area
	130	Block130 data area		130	Block130 data area
	131	Block131 data area		131	Block131 data area
	132	Block132 data area		132	Block132 data area
	133	Block133 data area		133	Block133 data area

ゾ ー ン 3	256	Block256 data area	ゾ ー ン 3	256	Block256 data area
	257	Block257 data area		257	Block257 data area
	258	Block258 data area		258	Block258 data area
	259	Block259 data area		259	Block259 data area
	260	Block260 data area		260	Block260 data area
	261	Block261 data area		261	Block261 data area
	262	Block262 data area		262	Block262 data area
NG ゾ ー ン 4	384	Block384 data area	NG ゾ ー ン 4	384	Block384 data area
	385	Block385 data area		385	Block385 data area
	386	Block386 data area		386	Block386 data area
	387	Block387 data area		387	Block387 data area
	388	Block388 data area		388	Block388 data area
	389	Block389 data area		389	Block389 data area

冗長ブロック			冗長ブロック		
			2	Block2 data area	
			5	Block5 data area	
			129	Block129 data area	
			131	Block131 data area	

【図 3 7】

ブロック置き換え前

物理ブロック
アドレス

NG	0	Block0 data area
	1	Block1 data area
NG	2	Block2 data area
	3	Block3 data area
	4	Block4 data area
NG	5	Block5 data area
NG	6	Block6 data area
	7	Block7 data area
	8	Block8 data area
NG	9	Block9 data area
	10	Block10 data area
NG	11	Block11 data area
	12	Block12 data area
	13	Block13 data area
	14	Block14 data area
	⋮	⋮
NG	256	Block256 data area
	257	Block257 data area
	258	Block258 data area
NG	259	Block259 data area
	260	Block260 data area
NG	261	Block261 data area
	262	Block262 data area
	263	Block263 data area
NG	264	Block264 data area
NG	265	Block265 data area
	266	Block266 data area
	267	Block267 data area
	268	Block268 data area
NG	269	Block269 data area
	⋮	⋮

冗長ブロック

ブロック置き換え後

物理ブロック
アドレス

	0	Block0 data area
	1	Block1 data area
	2	Block2 data area
	3	Block3 data area
	4	Block4 data area
	5	Block5 data area
	6	Block6 data area
	7	Block7 data area
	8	Block8 data area
	9	Block9 data area
	10	Block10 data area
	11	Block11 data area
	12	Block12 data area
	13	Block13 data area
	14	Block14 data area
	⋮	⋮
NG	256	Block256 data area
	257	Block257 data area
	258	Block258 data area
NG	259	Block259 data area
	260	Block260 data area
NG	261	Block261 data area
	262	Block262 data area
	263	Block263 data area
NG	264	Block264 data area
NG	265	Block265 data area
	266	Block266 data area
	267	Block267 data area
	268	Block268 data area
NG	269	Block269 data area
	⋮	⋮

冗長ブロックハハードウェアリダンダント

冗長ブロックハハードウェアリダンダント

冗長ブロックハハードウェアリダンダント

冗長ブロックハハードウェアリダンダント

冗長ブロックハハードウェアリダンダント

冗長ブロック

0	Block0 data area
2	Block2 data area
4	Block4 data area
5	Block5 data area
9	Block9 data area
11	Block11 data area

【図 3 8】

ブロック置き換え前

物理ブロック
アドレス

NG	ゾ ー ン 1	0	Block0 data area
		1	Block1 data area
		2	Block2 data area
		3	Block3 data area
		4	Block4 data area
		5	Block5 data area
		6	Block6 data area
	⋮	⋮	
NG NG NG	ゾ ー ン 2	128	Block128 data area
		129	Block129 data area
		130	Block130 data area
		131	Block131 data area
		132	Block132 data area
		133	Block133 data area
		⋮	⋮
	ゾ ー ン 3	256	Block256 data area
		257	Block257 data area
		258	Block258 data area
		259	Block259 data area
		260	Block260 data area
		261	Block261 data area
		262	Block262 data area
⋮	⋮		
NG NG	ゾ ー ン 4	384	Block384 data area
		385	Block385 data area
		386	Block386 data area
		387	Block387 data area
		388	Block388 data area
		389	Block389 data area
		⋮	⋮

ブロック置き換え後

物理ブロック アドレス

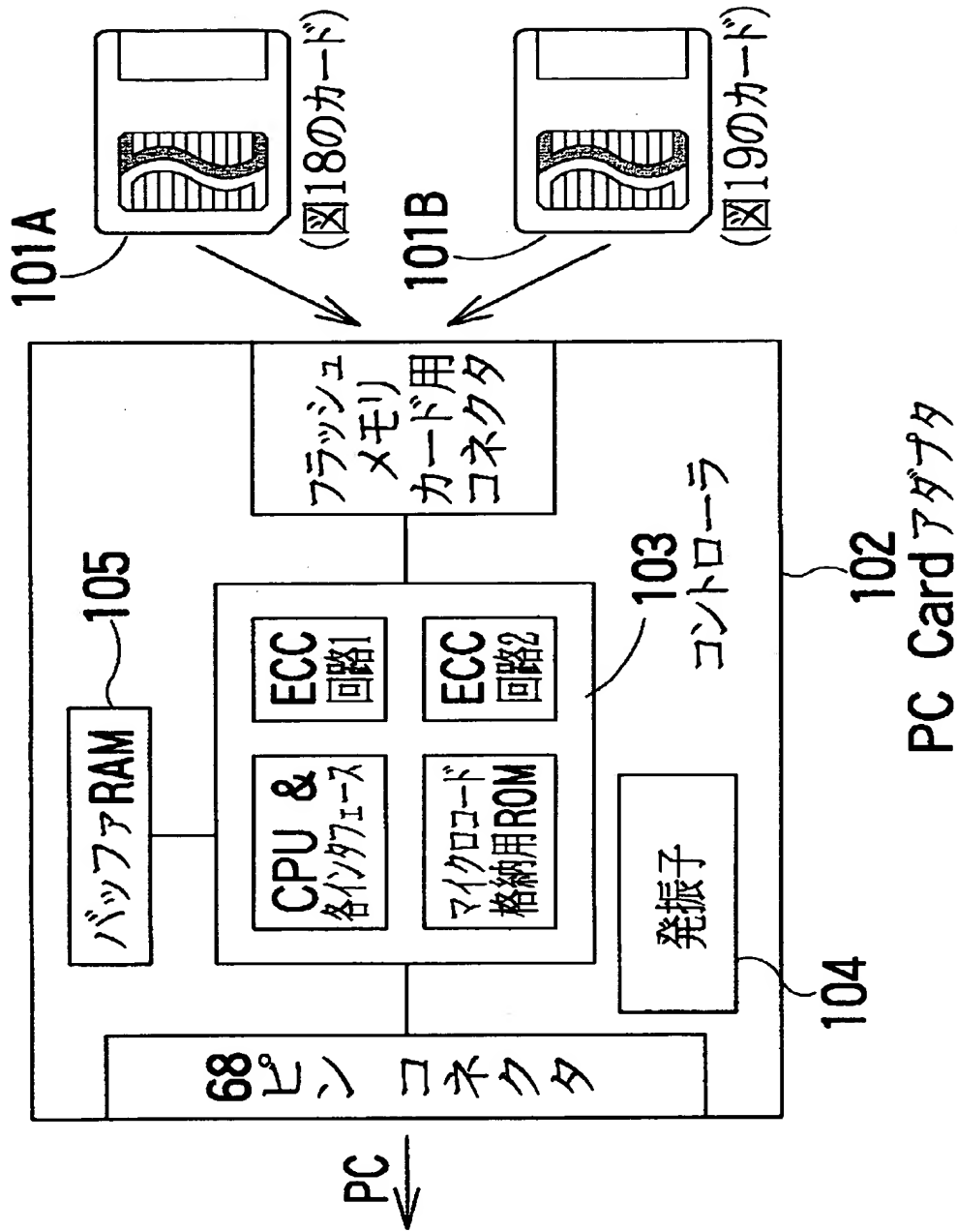
NG	ゾ ー ン 1	0	Block0 data area
		1	Block1 data area
		2	Block2 data area
		3	Block3 data area
		4	Block4 data area
		5	Block5 data area
		6	Block6 data area
		⋮	⋮
NG	ゾ ー ン 2	128	Block128 data area
		129	Block129 data area
		130	Block130 data area
		131	Block131 data area
		132	Block132 data area
		133	Block133 data area
		⋮	⋮
		NG	ゾ ー ン 3
257	Block257 data area		
258	Block258 data area		
259	Block259 data area		
260	Block260 data area		
261	Block261 data area		
262	Block262 data area		
⋮	⋮		
NG	ゾ ー ン 4	384	Block384 data area
		385	Block385 data area
		386	Block386 data area
		387	Block387 data area
		388	Block388 data area
		389	Block389 data area
		⋮	⋮
		⋮	⋮

冗長ブロック

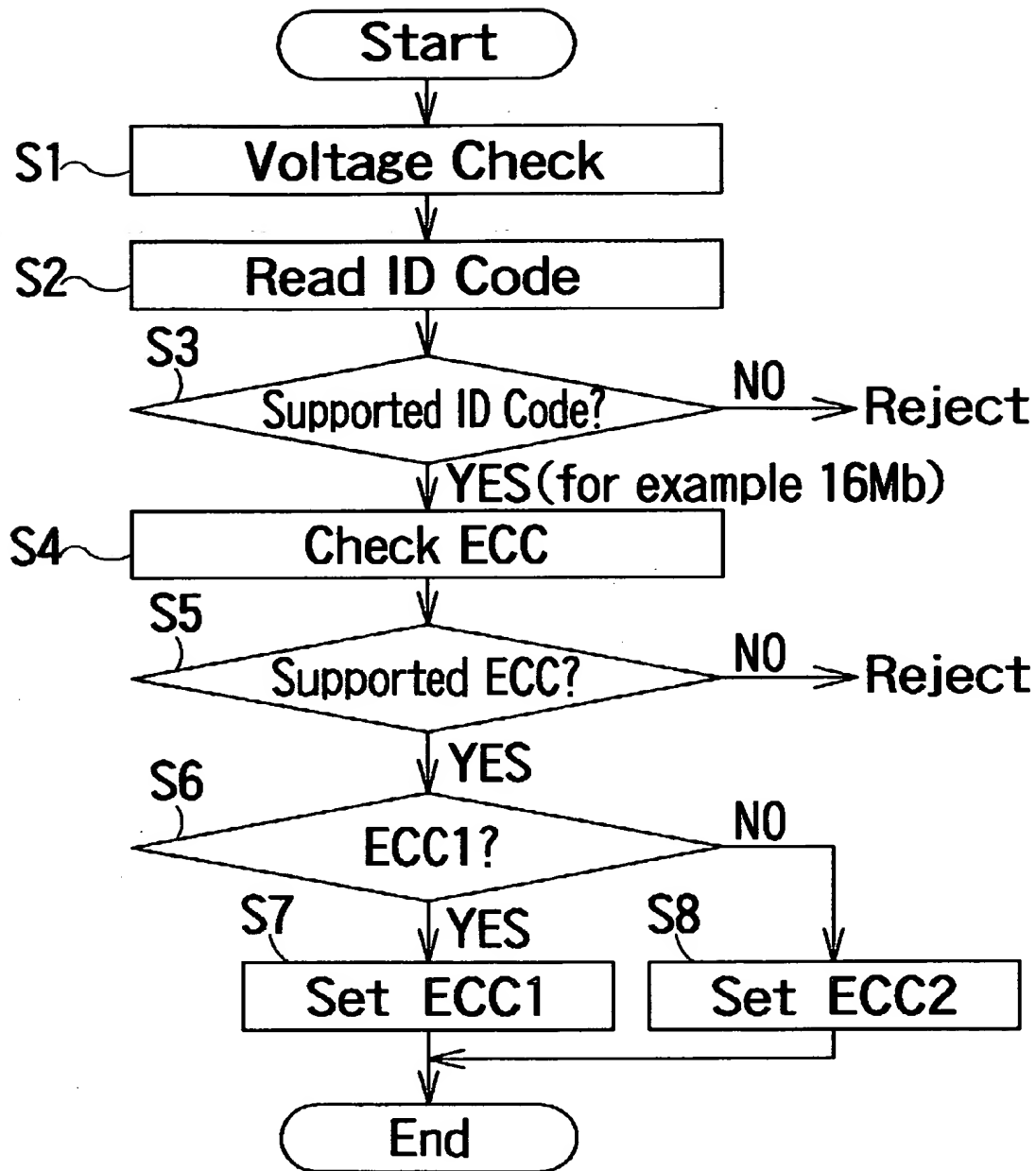
冗長ブロック

129	Block129 data area
131	Block131 data area
2	Block2 data area
385	Block385 data area

【図 39】



【図 40】



【図 4 1】

データ部

バイト	0ページ (偶数ページ)	1ページ (奇数ページ)
0~255	DATA Area-1	DATA Area-2

冗長部

バイト	偶数ページ	奇数ページ
256	ECC Flag Area	ECC Area-2
257	ECC Area-3	
258		
259		
260	Data Status Area	Block Address Area-2
261	Block Status Area	ECC Area-1
262	Block Address	
263	Area-1	

【図 4 2】

	ECC-Area1	ECC-Area2	ECC-Area3	ECC-Area4
ECC方式 1	Data Area-1 に対する ECC符号	Data Area-2 に対する ECC符号	空白 (ALL"FFh")	ECC1-Flag
ECC方式 2	Data Area-1,2 に対する ECC符号	Data Area-1,2 に対する ECC符号	Data Area-1,2 に対する ECC符号	ECC2-Flag

【書類名】 要約書

【要約】

【課題と解決手段】 システムが管理する論理ブロックと、複数のメモリセルにより構成され、前記論理ブロックに対応するデータを記憶する物理ブロックと、前記物理ブロック中に含まれ、前記対応する論理ブロックのアドレスを記憶する冗長部と、前記物理ブロックが少なくとも2以上で構成される物理ブロックエリアとを具備し、前記論理ブロックと前記物理ブロックエリアとの対応関係を管理するための論理番地／物理番地変換テーブルを作成する。

【選択図】 図33

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000003078

【住所又は居所】 神奈川県川崎市幸区堀川町72番地

【氏名又は名称】 株式会社東芝

【代理人】 申請人

【識別番号】 100064285

【住所又は居所】 東京都千代田区丸の内3-2-3 富士ビル 協和
特許法律事務所内

【氏名又は名称】 佐藤 一雄

【選任した代理人】

【識別番号】 100073379

【住所又は居所】 東京都千代田区丸の内3-2-3 富士ビル 協和
特許法律事務所

【氏名又は名称】 佐藤 政光

【選任した代理人】

【識別番号】 100088889

【住所又は居所】 東京都千代田区丸の内3丁目2番3号 協和特許法
律事務所

【氏名又は名称】 橘谷 英俊

【選任した代理人】

【識別番号】 100082991

【住所又は居所】 東京都千代田区丸の内3丁目2番3号 富士ビル
協和特許法律事務所

【氏名又は名称】 佐藤 泰和

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日
[変更理由] 新規登録
住 所 神奈川県川崎市幸区堀川町72番地
氏 名 株式会社東芝